

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
26 May 2006 (26.05.2006)

PCT

(10) International Publication Number
WO 2006/055445 A2

(51) International Patent Classification:
G06F 12/14 (2006.01)

(21) International Application Number:
PCT/US2005/041024

(22) International Filing Date:
14 November 2005 (14.11.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/627,502 13 November 2004 (13.11.2004) US
60/628,517 15 November 2004 (15.11.2004) US

(71) Applicant (for all designated States except US): **STREAM THEORY, INC.** [US/US]; Suite 350, 19600 Fairchild Road, Irvine, CA 92612 (US).

(72) Inventors: **DE VRIES, Jeffrey. ZIVERTNIK, Greg. HUBBLE, Ann.**

(74) Agent: **AHMANN, William**; Perkins Coie LLP, 101 Jefferson Drive, Menlo Park, CA 94025 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: **HYBRID LOCAL/REMOTE STREAMING**

(57) Abstract: A technique for streaming from a local device and a remote device involves providing a subset of data associated with a streaming application to a local device. An example of a method according to the technique includes running a streaming software player, accessing data from a local storage device necessary to stream a software application, and accessing data from a remote location necessary to stream the software application using the streaming software player. An example of a system according to the technique may include, by way of example but not limitation, a means for streaming a first subset of content associated with a streaming application from a streaming server, a means for streaming a second subset of content associated with the streaming application from a local source, and a means for periodically querying the streaming server, wherein the periodic querying informs the streaming server that the streaming application is running.

WO 2006/055445 A2

HYBRID LOCAL/REMOTE STREAMING

BACKGROUND

Software streaming involves downloading small pieces of files as the pieces are needed by the program being streamed. These small pieces may be referred to as blocks. A streaming client sends requests for blocks as they are needed up to a streaming server, which sends back streaming data that is associated with the requested block. Sending a request and receiving the streaming data may cause delays that can slow down the streamed program.

One advantage of streamed applications is that users only download what is needed to execute a program. The provider of a streamed application has more control over the streamed application than, for example, a provider of a program that is installed to hard disk. For example, the provider of a streamed application may charge a per hour rate for the use of the application. Alternatively, the provider of a streamed application may prevent a local disk from being filled up unnecessarily with code that will not be accessed.

While streaming has certain advantages, there are many problems associated with streaming software that it would be advantageous to negate, work around, or reduce. For example, speeding up execution of streaming programs is an on-going issue. As another example, it may be desirable to provide a user with an experience similar to that associated with an installed, as opposed to streamed, program. As another example, it may be desirable to provide streamed programs on consoles.

In addition, a provider of a streamed application may desire to attempt copyright protection. This can be difficult because if a streamed application is downloaded entirely to a local machine, the user of the local machine may gain access to all portions of the streamed application, eliminating the need to stream and increasing the risks of copyright violations. Thus, an additional concern for some may be to prevent a user from gaining access to all of a given program locally.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention are illustrated in the figures. However, the embodiments and figures are illustrative rather than limiting; they provide examples of the invention.

FIG. 1 depicts a flowchart of a process for streaming a software title and other activities associated with the software title.

FIGS. 2A to 2N are intended to illustrate an example of a streaming software experience.

FIGS. 3A and 3B depict systems for hybrid local/remote streaming.

FIGS. 4A and 4B depict examples of a system for hybrid local/networked streaming.

FIG. 5 depicts a flowchart of a method for hybrid local/remote software streaming.

FIG. 6 depicts a flowchart of a method for hybrid local/remote software streaming.

FIG. 7 depicts a networked system for use in an embodiment.

FIG. 8 depicts a computer system for use in the system of FIG. 7.

files, etc.) These can be opened modified, copied, or inserted locally. Similarly, users may be able to download a mod from a network and integrate the mod into the software title.

Users and software title providers typically also find software patches desirable. A streamed application may provide a method for users to download the latest patches.

5 As is common for most software titles in the industry, product registration may be desired when providing CD or DVD streaming titles.

Compression techniques may be applied to fully utilize media on which streamed data files are provided. For example, packaging of data into streaming buffers may be as compact as Zip-style compression.

10 FIG. 1 depicts a flowchart of a typical process for streaming a software title, and other activities associated with the software title. In a non-limiting embodiment, at module 102 a user puts a CD into a computer, as illustrated in FIG. 2A. In a non-limiting embodiment, upon putting the CD into the computer, an "Autorun.inf" file is processed by the operating system (e.g., Windows 2000) of the computer. The Autorun.inf file references and starts a "Stub"
15 program. At module 104, the Stub program provides initial options to the user.

The Stub program may provide features, such as product registration, streaming player (e.g., Stream Weaver) detection, streaming player installation, providing the user with various menu options, etc. Product registration may involve determining whether a CD registration key has been entered by checking persistent memory on the user's computer. In an embodiment, if
20 the product has not been registered (typically, this is true the first time a title is streamed), then a registration program prompts the user for the relevant registration information, as illustrated in FIG. 2B.

In an embodiment, the Stub program may provide a screen for entering the CD registration key and product registration. The Stub program may save information about the CD registration key and product registration in persistent memory. Registration information may be
25 sent to a vendor or producer of the software title over a network (e.g., the Internet). If no network connection is available, registration may be handled in some other manner, as is well-understood in the art of software registration.

If the user skips the registration process, this fact may be saved in persistent memory, and
30 used when deciding whether to ask the user to register again later. If the registration has been

used to check keys or other data, as necessary. At optional module 120 (FIG. 1) the executable checks Macrovision Safedisc. The file system must be intelligent enough to identify on which disc desired files are located, prompting the user to swap discs if necessary at module 122 (FIG. 1). If installed to hard drive, when the file system requests another CD, the user may, for example, enter a key sequence to go on. In this example, the key sequence may be intercepted by a driver that lets the game proceed as if another disc was inserted. Other techniques may also be used to provide multi-disc functionality.

At module 124 (FIG. 1), if the user clicks on the install-to-hard-drive option, as illustrated in FIG. 2H, the user is given the opportunity to install the streaming application onto a (usually) local hard drive. The techniques involved in handling an installed streaming application and a streaming application running from multiple CDs are similar. Each CD is equivalent to a stream file. When the application prompts for a particular CD, this is detected by requests for blocks outside the range of the currently active stream file. The streaming player would have to intercept and detect requests to change CDs.

When installing a streaming title on the hard drive, the system typically checks for adequate disk space and copies all of the streaming files to the hard drive, if space is sufficient, as illustrated in FIG. 2I. In addition, shortcuts may be created at optional module 126 (FIG. 1), window registry settings may be adjusted, and removal code for the short cuts may be added to the "Add and Remove" programs. Prompting of the user to load the CDs in a certain order may be required. Once a streaming title has been installed onto the hard drive, the user may select, for example, a shortcut to launch the streaming player and run the streaming application, as illustrated in FIG. 2J. The next time "Play Now" is selected, the cache on the hard drive can be used, as indicated at module 128 (FIG. 1).

At module 112 (FIG. 1), if the user clicks on the file-editor option, as illustrated in FIG. 2K, a file editor is launched at module 130 (FIG. 1), as illustrated in FIG. 2L. The user may gain access to specific files that can be modified or replaced. Normally, the file structure of a streamed application is hidden from the user. If an application modifies a file, this file is copied out a subdirectory, such as the directory "C:\Stream Theory\local\app_name\...\file_to_be_edited".

As with application patching, the editor should be processed with the streamed application and included in the same stream file as the streamed application so that the files are

storage, such as a hard disk drive. The removable media may include copy protection safeguards of types that are well-known in the art and are continuously being developed to protect against privacy and other forms of intellectual property theft.

5 In an embodiment, the second streaming content database 304 does not include all of the content associated with a streaming application. The first streaming content database 302, however, includes the rest of the streaming content for the streaming application. As is known in the art of content streaming, the local cache 306 may be utilized while the streaming player 308 runs the streaming application. When content, such as a block, is requested from the first streaming content, the streaming player requests the block from the streaming server 310. On
10 the other hand, when content, such as another block, is requested from the second streaming content, the streaming player requests the block from a local source, such as the removable media on which the second streaming content database 304 was provided, or a local source, such as a hard drive onto which the second streaming content database 304 was installed. A request may be honored by the local cache 306 if the requested content, such as a block, is cached.

15 Advantageously, the system 300 can be used to stream software at a high rate of speed, even if broadband access is unavailable or limited, because some of the requested blocks are available locally. For large software titles, this can be of great value. Moreover, it is sometimes valuable to keep certain data relatively secure. The data that should be kept secure can be encoded into the blocks that are provided locally.

20 FIG. 3B depicts an example of the system 300 that includes a heartbeat monitor 330. The heartbeat monitor 330 is occasionally pinged, queried, or otherwise notified of the ongoing streaming of a streaming application. It may be desirable to know how much a software title is used, and for how long, for the purposes of, for example, charging for the time spent using the software title. The heartbeat monitor 330 may receive additional information, as well, such as
25 which blocks are being requested and in what order, that may help to predictively provide replies to block requests or to configure content databases in future versions of the software title.

FIG. 4A depicts an example of a system 400 for hybrid local/networked streaming. The system 400 includes a streaming player 402, a first content database 404, a second content database 406, and a network interface 408. The streaming player 402 and the first content
30 database 404 are on a streaming client 412, which is coupled to a network 420, which may

flowchart 500 continues with streaming 504 a second subset of content associated with the streaming application from a local source. In an embodiment, the flowchart 500 continues with periodically querying 506 the streaming server, wherein the periodic querying informs the streaming server that the streaming application is running.

5 FIG. 6 depicts a flowchart 600 of a method for hybrid local/remote software streaming. In an embodiment, the flowchart 600 starts with dividing 602 content associated with a streaming application into first content and second content. This may be conducted in a streamification stage of streaming software development. Alternatively, a previously streamified application may be divided according to an algorithm (e.g., removing every 20th block from a
10 first streaming content database and placing every 20th block in a second streaming content database). In an embodiment, the flowchart 600 continues with maintaining 604 the first content on a streaming server. This may entail providing a web site to which users may visit to begin streaming an application. The web site may be at the same physical location of the streaming content, or a remote location. In an embodiment, the flowchart 600 continues with receiving 606
15 a request to stream the streaming application. In an embodiment, the flowchart 600 continues with providing 608 a token file in response to the request, the token file including:

- an address associated with the streaming server;
- instructions to receive the first content from the streaming server; and
- instructions to receive the second content from a local streaming source.

20 The streaming software player used to stream applications, in an embodiment, may make use of a DLL that is used to determine if an FSD has been loaded or if a user has been loaded. The user should be able to stream from multiple stream files. Common files of an application must fit on one CD. The user should be able to intercept requests for new discs or the virtual
25 equivalent for an installed to hard drive software title switching stream files.

 For files installed on the hard drive, the registry should be checked for the streaming software player to determine if streaming is to be done from a CD, from the stream files stored on the hard disk, or from somewhere else. If it is desired for a streaming application to run like software that has been fully installed on disk, then that streaming application should know when
30 it is to switch to a different stream file. The user file system driver may support a system level call to switch between stream files for this purpose. Fully installed applications, may make an

This causes a problem if the application tries to reference a part of the application that is not present because the CD with the stream data blocks is not present. Even though the read file appears to be available, the read will fail. Because of this, the application has this structural dichotomy that it appears as if all the files are present but the application has to be aware of when it needs to request that a different CD be inserted. Like in a non-streamed multi-CD game, the game puts up a request for the new CD to be inserted. This also implies that each CD can't have an overlapping directory structure.

Streamed applications may be structured so that the parts of the application that are common between CDs are known. The streamed application can lock these common parts into the cache or copy them to the hard drive. These files may be identified at the time the application is turned into a streamed application.

Online game play may proceed similarly to a game accessed through a CD or through a CD with formatted stream data. The access time to retrieve executables or data for a game that was traditionally executed from a CD is virtually identical to the time required to access data from a CD with stream data. This is also true for applications saved to disk. Additionally, given in this situation that none of the executables or data are accessed over the network, there is no or minimal competition for bandwidth between on-line playing data exchanges and the accesses to the hard disk or CD.

The game file system may be modified to search the hierarchy of local, hard drive, and streamed CD caches. Files may have a disc (cache) identifier. The file system may prompt the user to insert the correct disc, if necessary. The game should limit disc swapping points for simplicity; this is likely on level boundaries (at least), so the swapping happens in the user interface shell, not the running gameplay. This directory could be pre-computed before hand and be an additional data file on the CD. Handling multiple discs will require duplicated data on each disc; this could impact overall media space requirements negatively. Possible alternatives to address this are to always do a partial install to the hard disc, installing shared components rather than duplicating them in the stream theory buffers on separate discs.

The following description of FIGS. 7 and 8 is intended to provide an overview of computer hardware and other operating components suitable for performing the methods of the invention described herein, but is not intended to limit the applicable environments. Similarly, the computer hardware and other operating components may be suitable as part of the

Client computer systems 712, 718, 722, and 726 can each, with the appropriate web browsing software, view HTML pages provided by the web server 704. The ISP 710 provides Internet connectivity to the client computer system 712 through the modem interface 714, which can be considered part of the client computer system 712. The client computer system can be a personal computer system, a network computer, a web TV system, or other computer system. While FIG. 7 shows the modem interface 714 generically as a "modem," the interface can be an analog modem, isdn modem, cable modem, satellite transmission interface (e.g. "direct PC"), or other interface for coupling a computer system to other computer systems.

Similar to the ISP 714, the ISP 716 provides Internet connectivity for client systems 718, 722, and 726, although as shown in FIG. 7, the connections are not the same for these three computer systems. Client computer system 718 is coupled through a modem interface 720 while client computer systems 722 and 726 are part of a LAN 730.

Client computer systems 722 and 726 are coupled to the LAN 730 through network interfaces 724 and 728, which can be Ethernet network or other network interfaces. The LAN 730 is also coupled to a gateway computer system 732 which can provide firewall and other Internet-related services for the local area network. This gateway computer system 732 is coupled to the ISP 716 to provide Internet connectivity to the client computer systems 722 and 726. The gateway computer system 732 can be a conventional server computer system.

Alternatively, a server computer system 734 can be directly coupled to the LAN 730 through a network interface 736 to provide files 738 and other services to the clients 722 and 726, without the need to connect to the Internet through the gateway system 732.

FIG. 8 depicts a computer system 740 for use in the system 700 (FIG. 7). The computer system 740 may be a conventional computer system that can be used as a client computer system or a server computer system or as a web server system. Such a computer system can be used to perform many of the functions of an Internet service provider, such as ISP 710 (FIG. 7).

In the example of FIG. 8, the computer system 740 includes a computer 742, I/O devices 744, and a display device 746. The computer 742 includes a processor 748, a communications interface 750, memory 752, display controller 754, non-volatile storage 756, and I/O controller 758. The computer system 740 may be couple to or include the I/O devices 744 and display device 746.

are connected together through bridge components that perform any necessary translation due to differing bus protocols.

Network computers are another type of computer system that can be used with the present invention. Network computers do not usually include a hard disk or other mass storage, and the executable programs are loaded from a network connection into the memory 752 for execution by the processor 748. A Web TV system, which is known in the art, is also considered to be a computer system according to the present invention, but it may lack some of the features shown in FIG. 8, such as certain input or output devices. A typical computer system will usually include at least a processor, memory, and a bus coupling the memory to the processor.

In addition, the computer system 740 is controlled by operating system software which includes a file management system, such as a disk operating system, which is part of the operating system software. One example of an operating system software with its associated file management system software is the family of operating systems known as Windows® from Microsoft Corporation of Redmond, Washington, and their associated file management systems. Another example of operating system software with its associated file management system software is the Linux operating system and its associated file management system. The file management system is typically stored in the non-volatile storage 756 and causes the processor 748 to execute the various acts required by the operating system to input and output data and to store data in memory, including storing files on the non-volatile storage 756.

Some portions of the detailed description are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of operations leading to a desired result. The operations are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to

CLAIMS

What is claimed is:

1. A method, comprising:
running a streaming software player;
5 accessing data from a local storage device necessary to stream a software application using the streaming software player;
accessing data from a local cache necessary to stream the software application using the streaming software player; and
accessing data from a remote location necessary to stream the software application using
10 the streaming software player.
2. The method of claim 1, further comprising sending a request to stream the software application.
3. The method of claim 1, further comprising receiving in response to a request to stream the software application a token file.
- 15 4. The method of claim 1, further comprising receiving in response to a request to stream the software application an address of a streaming server at the remote location.
5. The method of claim 1, further comprising receiving in response to a request to stream the software application instructions to access first content from the remote location and to access second content from the local storage device.
- 20 6. The method of claim 1, further comprising periodically querying a streaming server at the remote location, wherein the periodic querying informs the streaming server that the streaming application is running.
7. The method of claim 1, wherein the local storage device is a removable storage device.

14. A system, comprising:
a means for streaming a first subset of content associated with a streaming application from a streaming server;
a means for streaming a second subset of content associated with the streaming application from a local source; and
a means for periodically querying the streaming server, wherein the periodic querying informs the streaming server that the streaming application is running.
15. The system of claim 14, further comprising a means for running a streaming software player.
16. The system of claim 14, further comprising a means for sending a request to stream the software application to the streaming server.
17. The system of claim 14, further comprising a means for accessing data from a local storage device necessary to stream a software application.
18. The system of claim 14, further comprising a means for accessing data from a local cache necessary to stream the software application.
19. The system of claim 14, further comprising a means for accessing data from a remote location necessary to stream the software application using the streaming software player.
20. The system of claim 14, wherein the local source is a removable storage device.

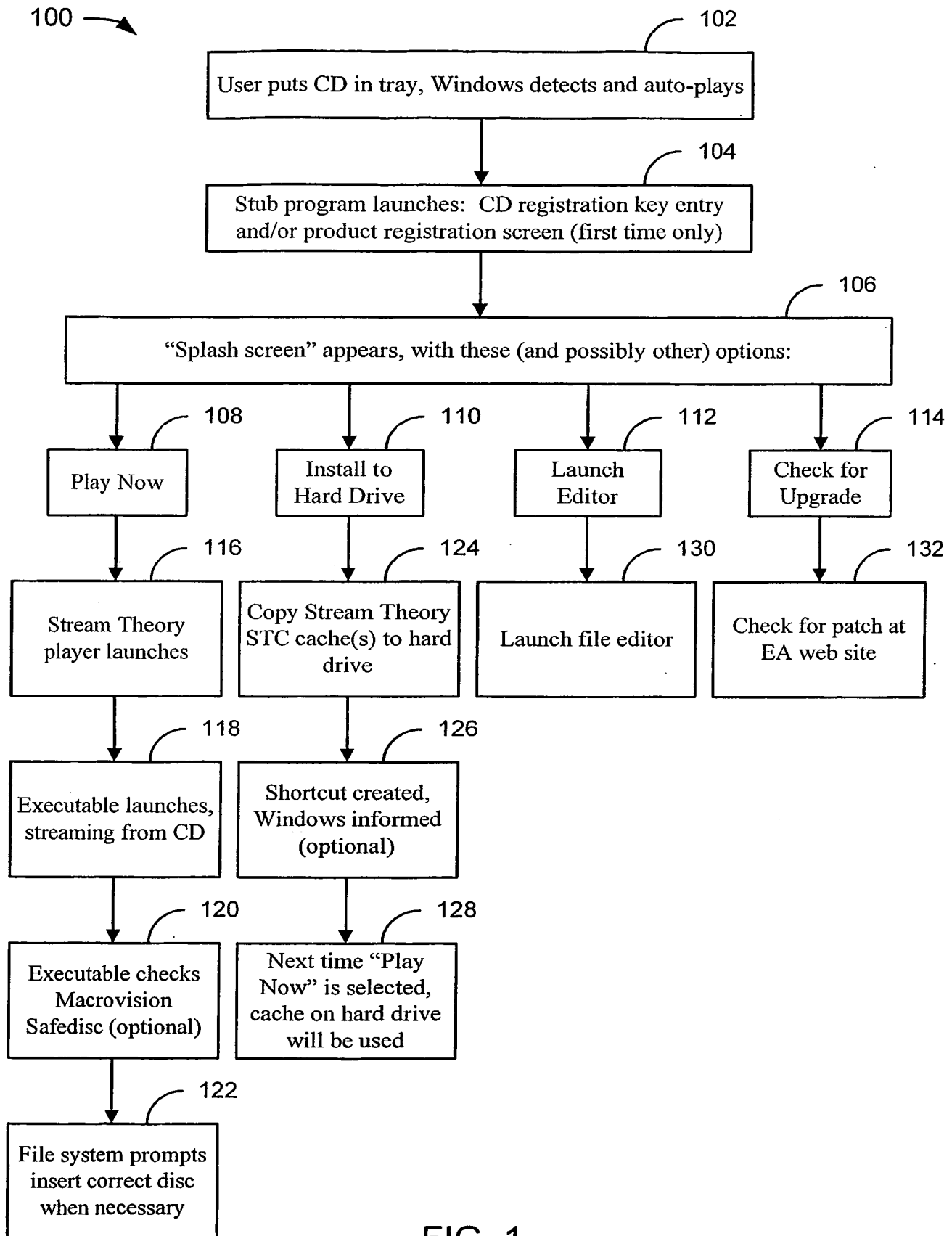


FIG. 1

The CD Streaming Experience

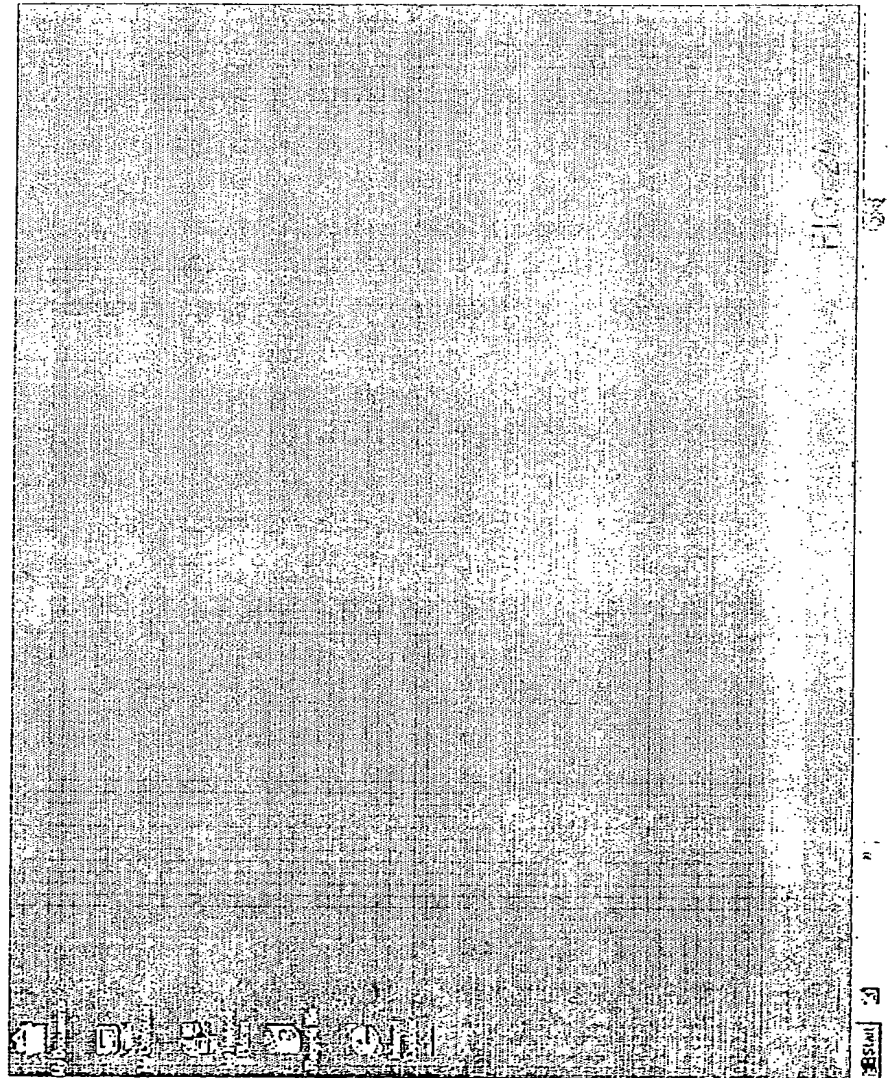
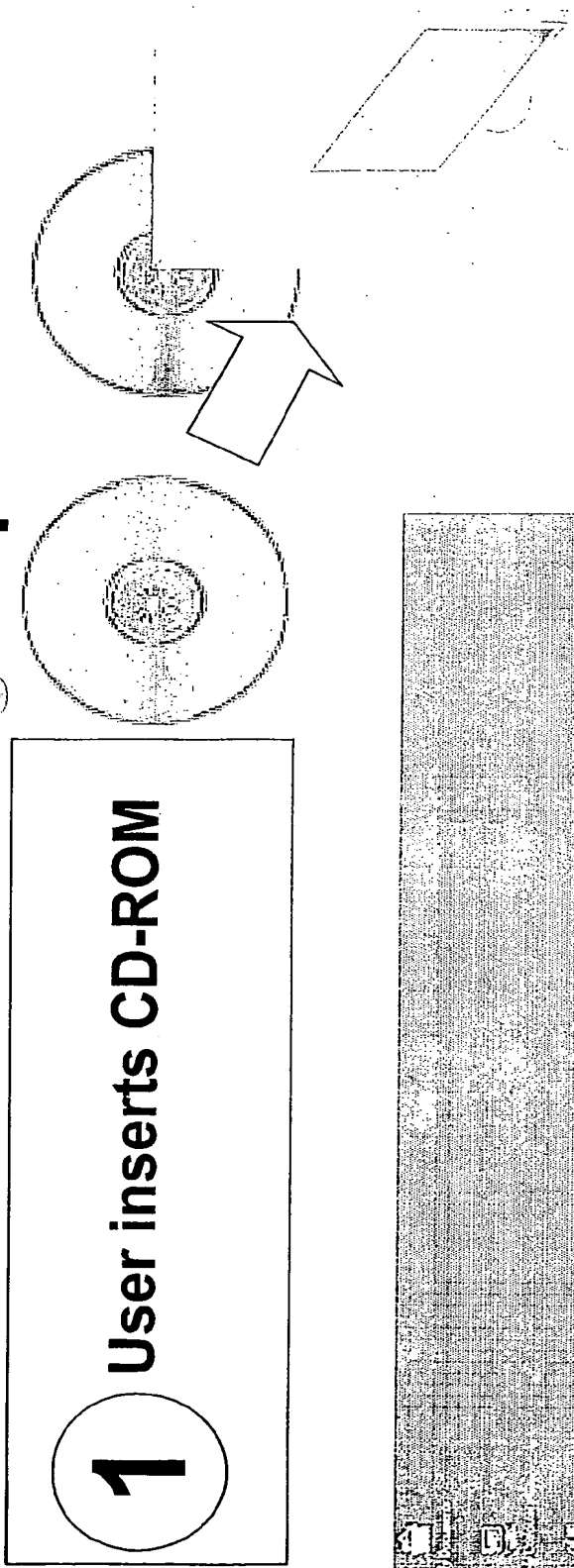


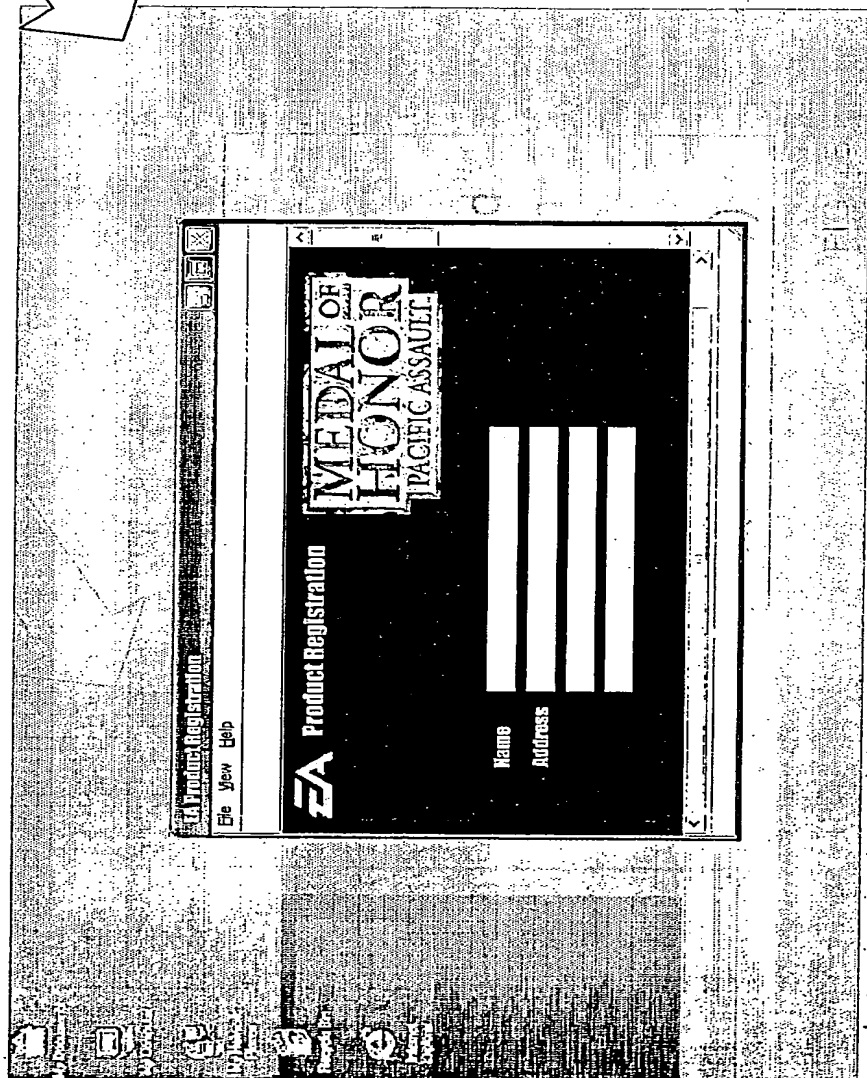
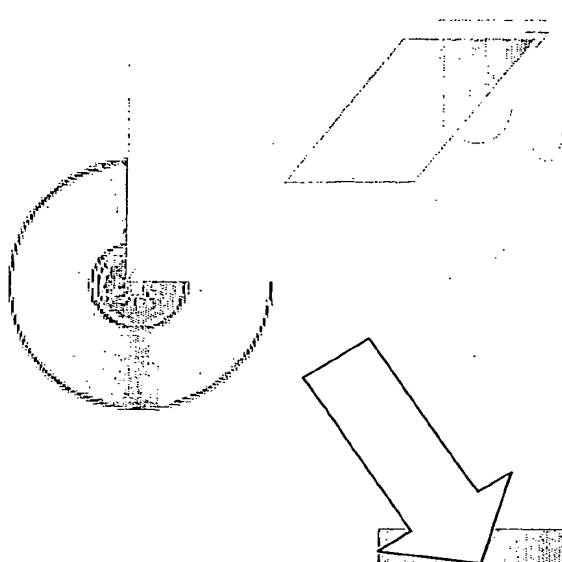
FIG. 2A

The CD Streaming Experience

WO 2006/055445

3/22

PCT/US2005/041024



2

CD autoruns and
EA's registration
program runs
automatically
(first time only)

FIG. 2B

The CD Streaming Experience

3

Stream Theory Player gets
automatically installed (if not
already installed)

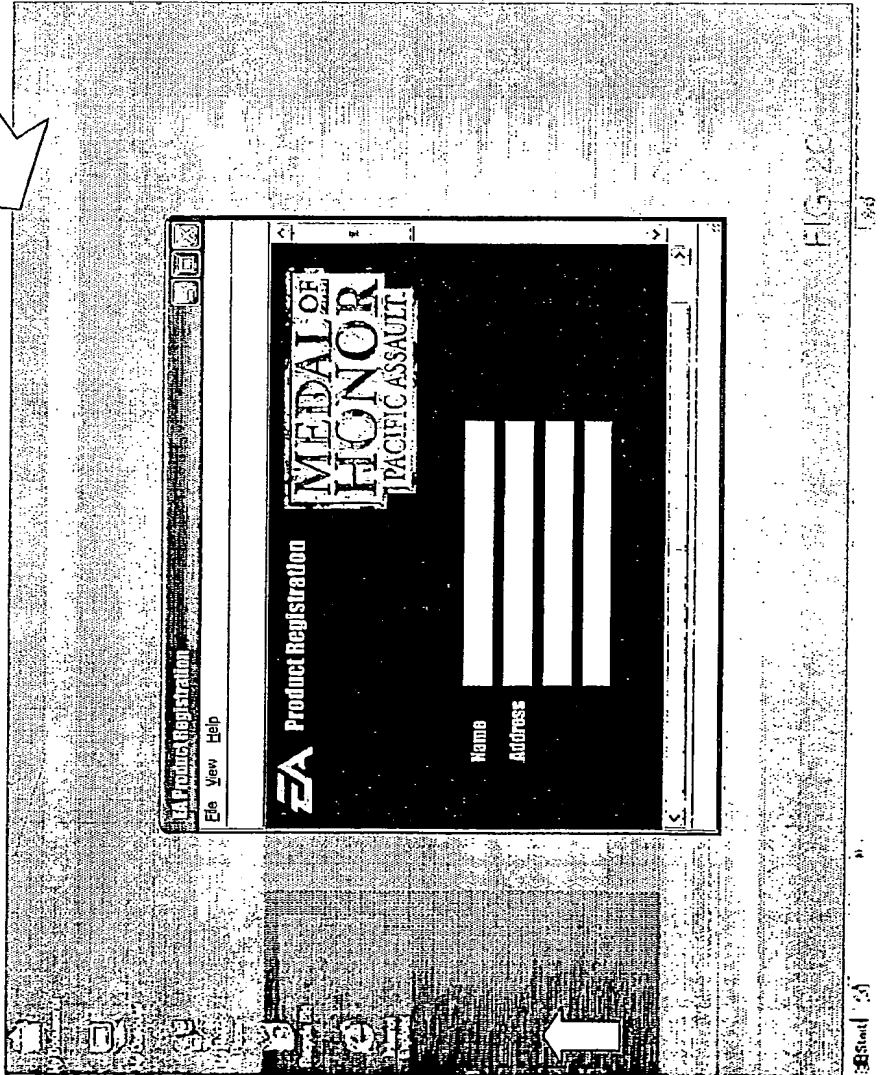
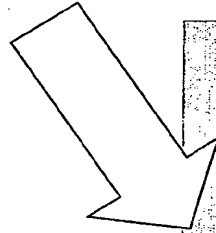
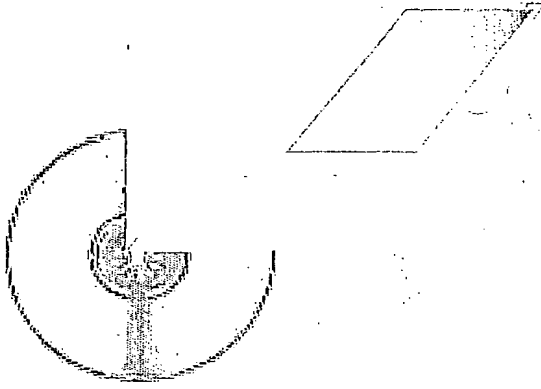


FIG. 2C

The CD Streaming Experience

4

Splash Screen is
Shown

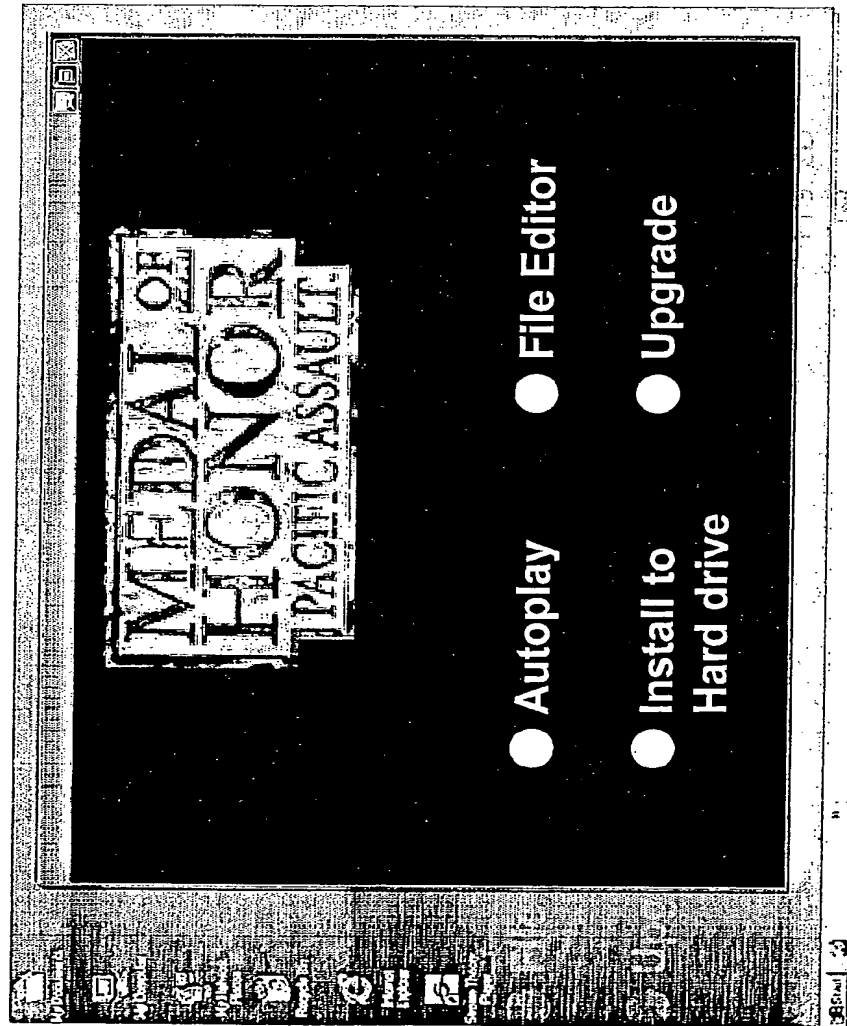
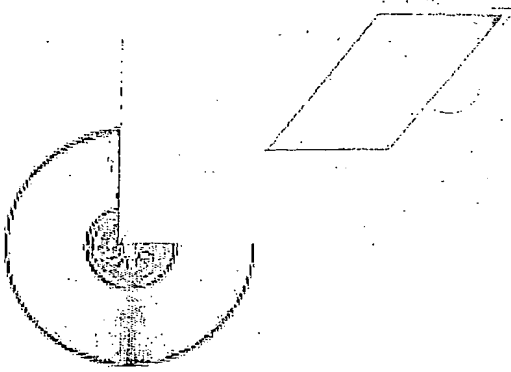


FIG. 2D

The CD Streaming Experience

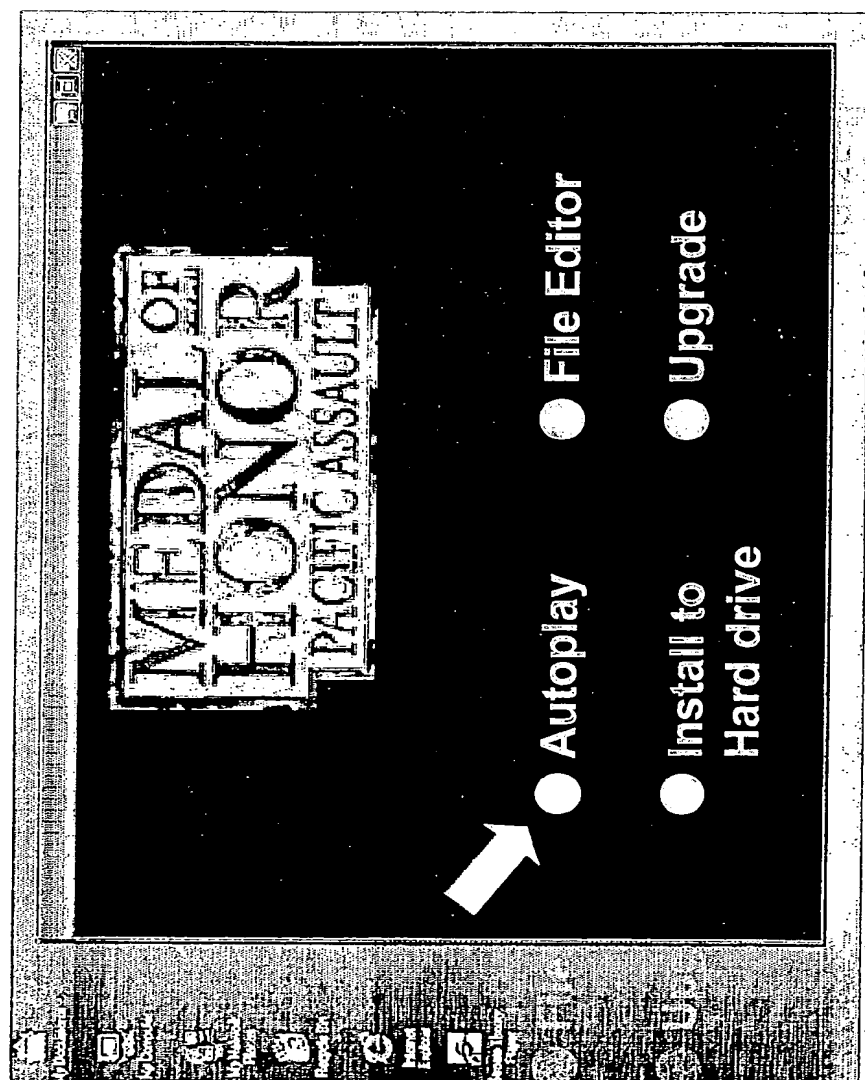
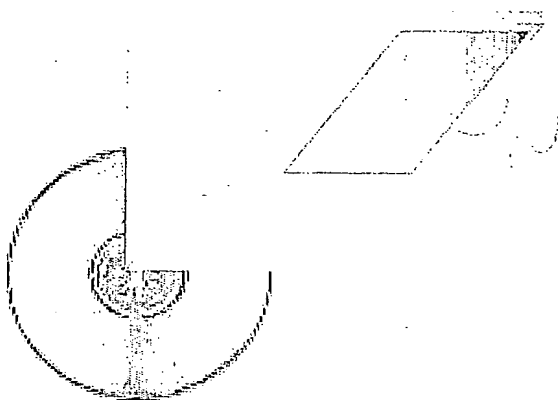
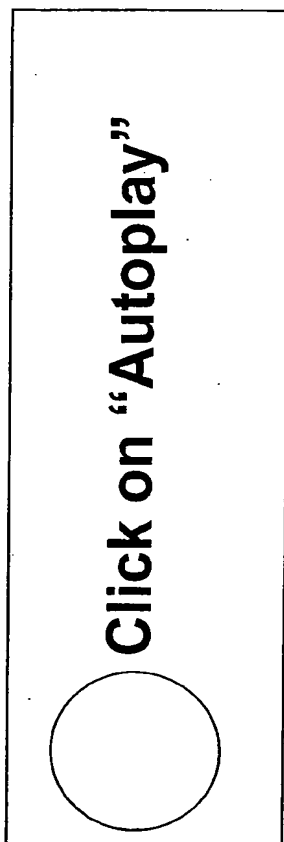
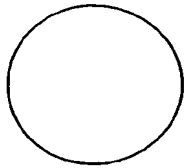


FIG. 2E

The CD Streaming Experience



Click on "Autoplay"

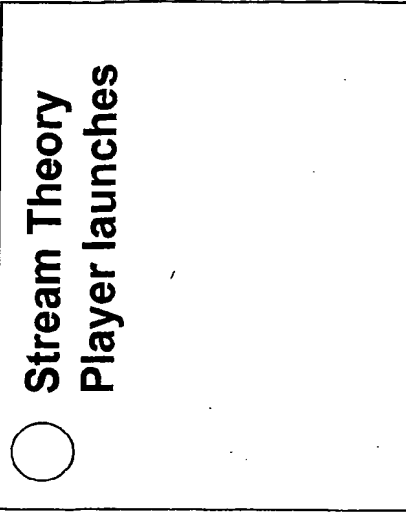
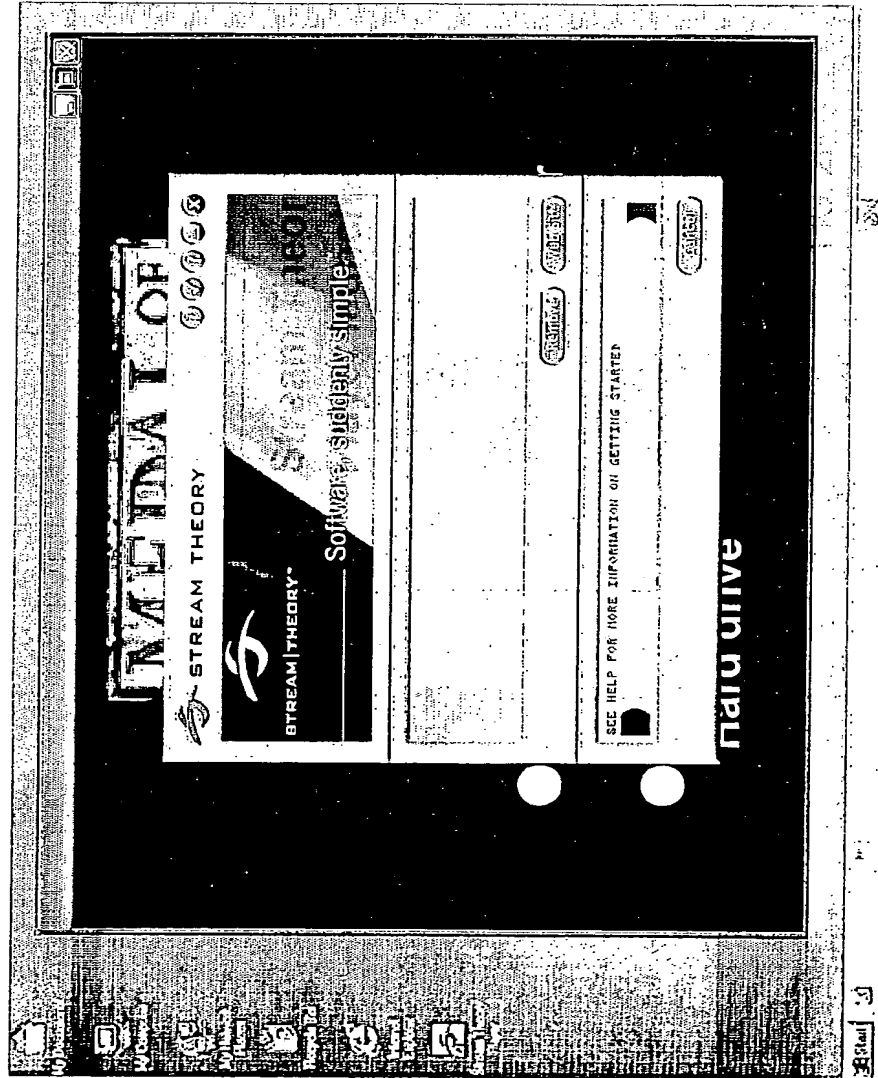
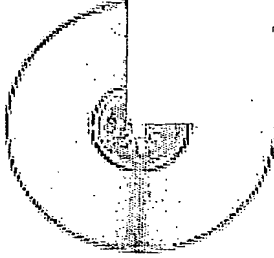
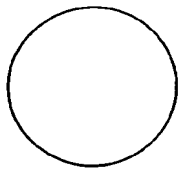
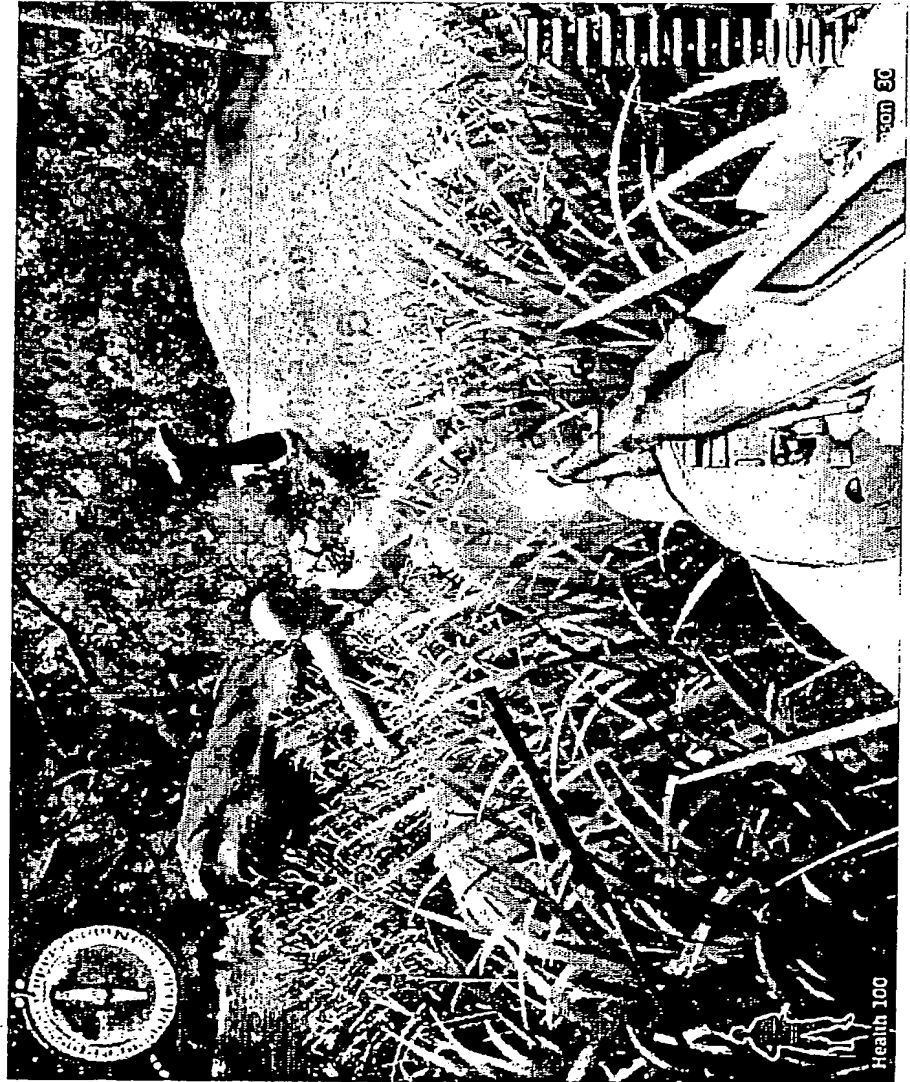
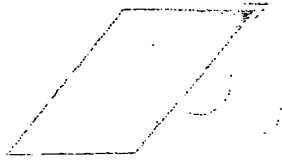
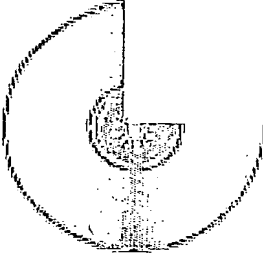


FIG. 2F

The CD Streaming Experience



Click on "Autoplay"



Game streams
from CD



Game plays using
data from CD and
"normal" Player
cache

FIG. 2G

The CD Streaming Experience

WO 2006/055445

9/22

PCT/US2005/041024

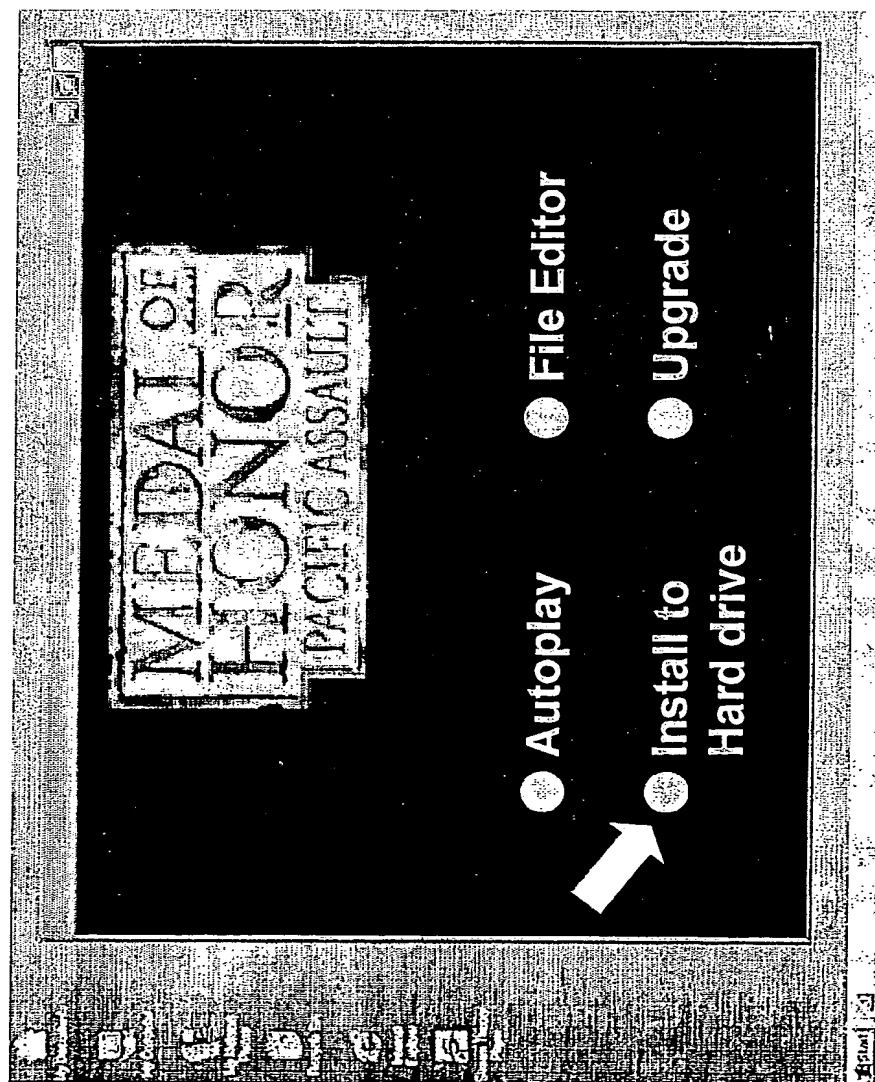
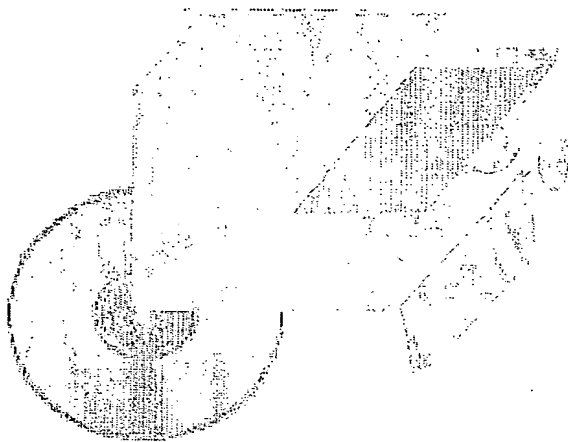
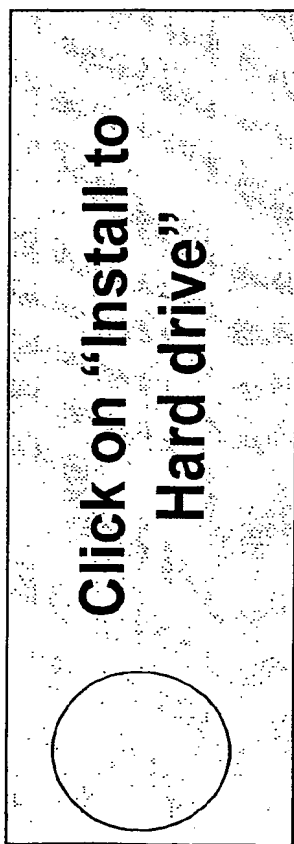


FIG. 2H

The CD Streaming Experience

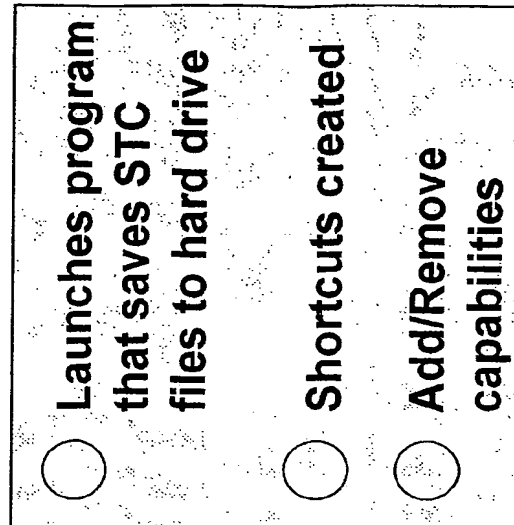
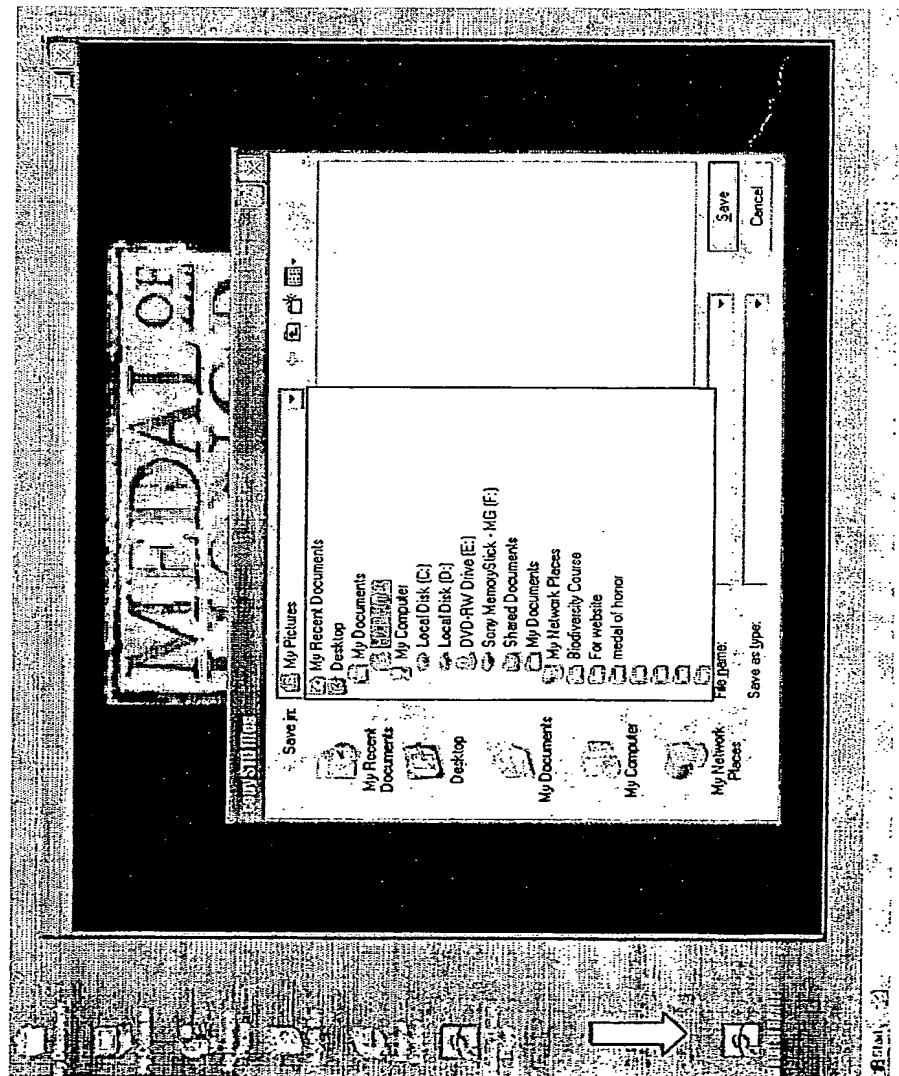


FIG. 2I

11/22

The CD Streaming Experience

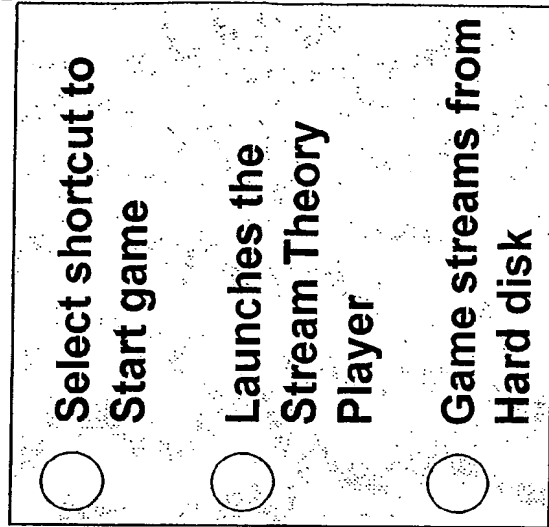
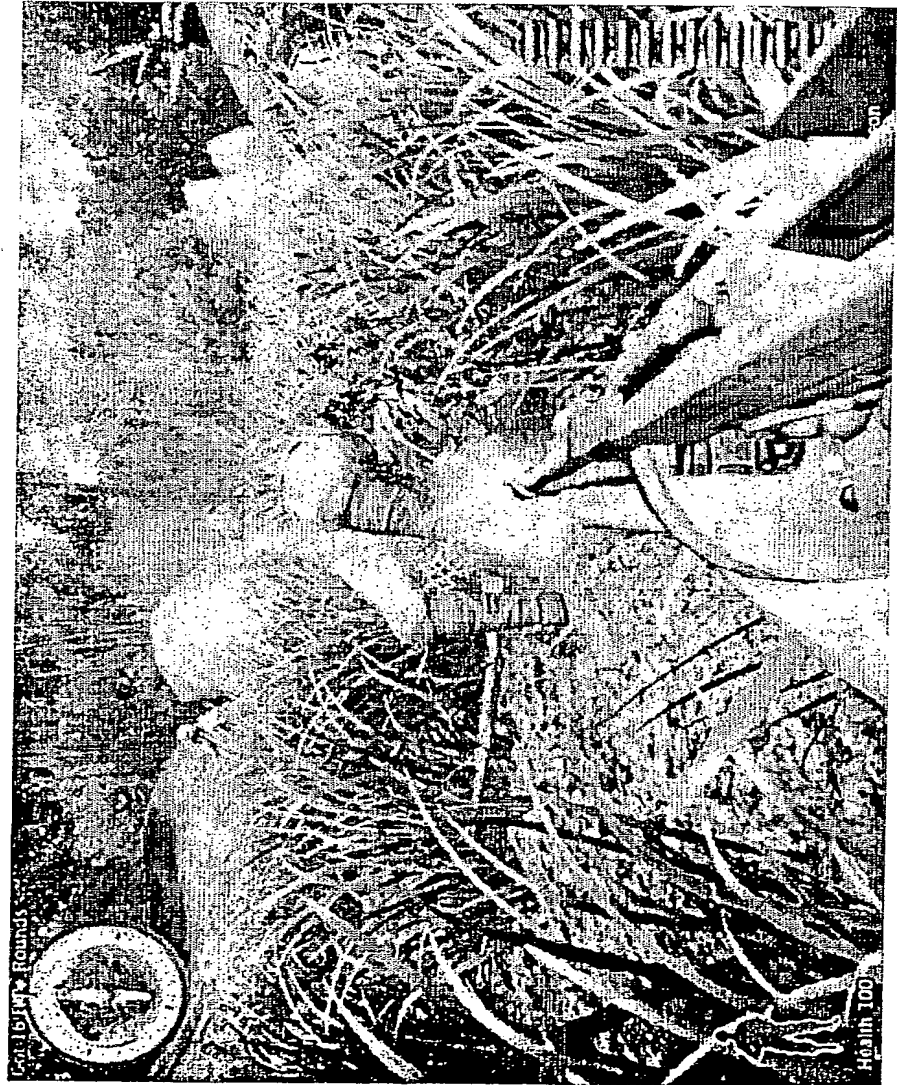
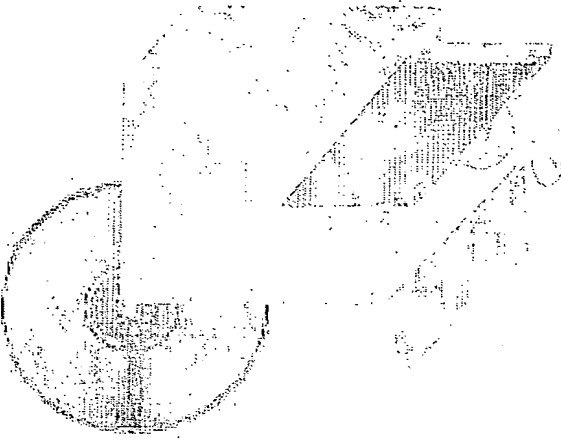


FIG. 2J

The CD Streaming Experience

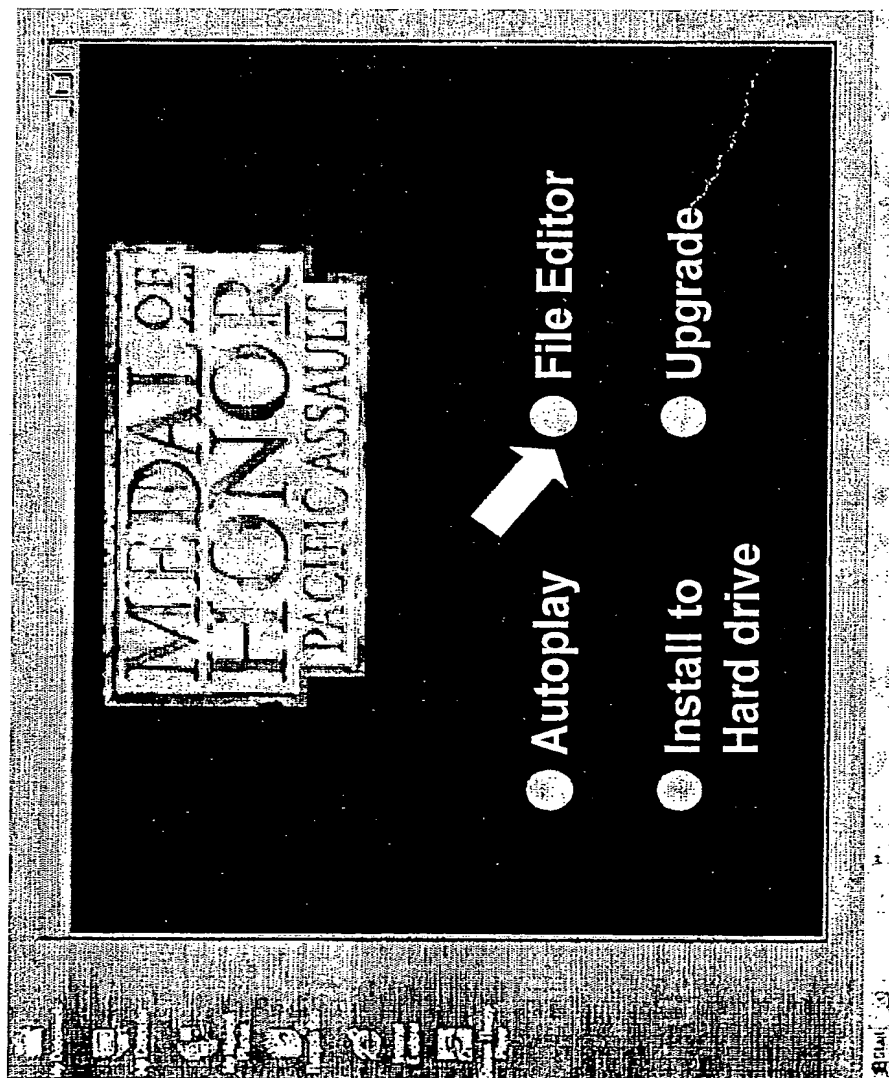
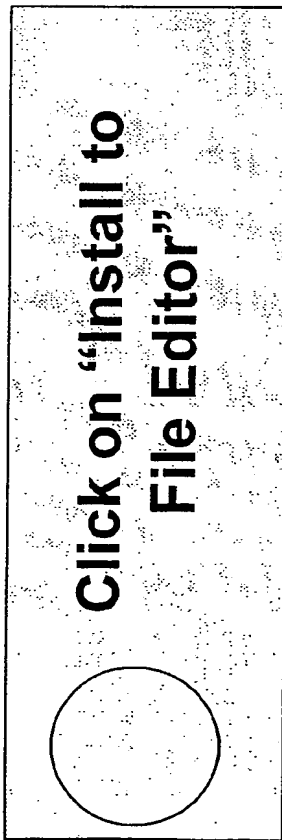


FIG. 2K

The CD Streaming Experience



WO 2006/055445

13/22

PCT/US2005/041024

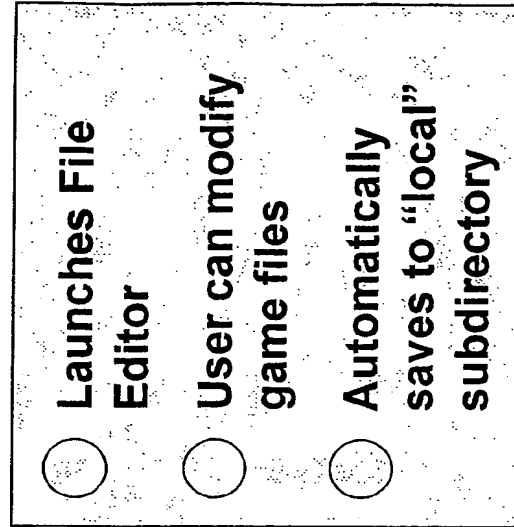
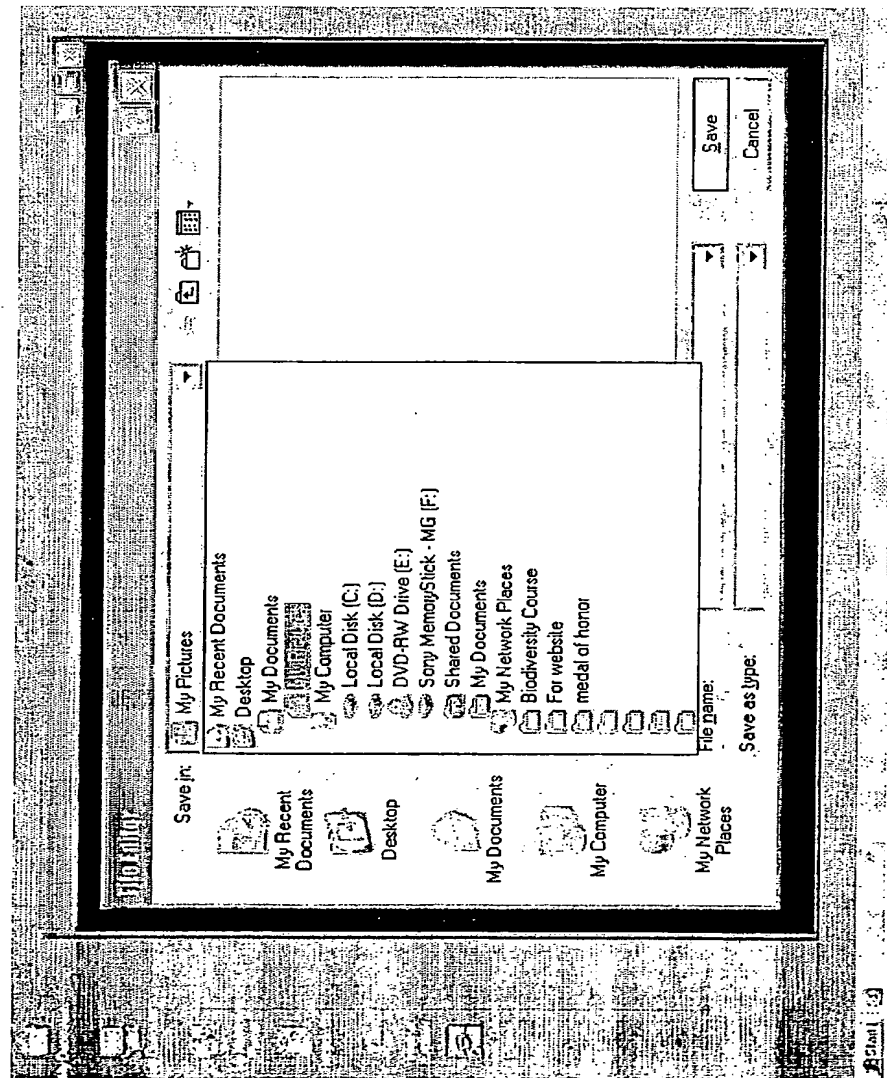
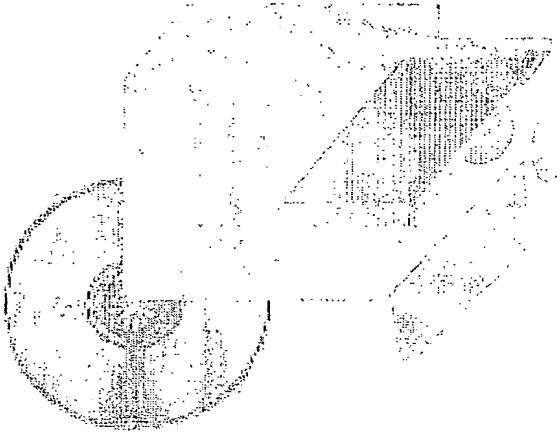


FIG. 2L

The CD Streaming Experience

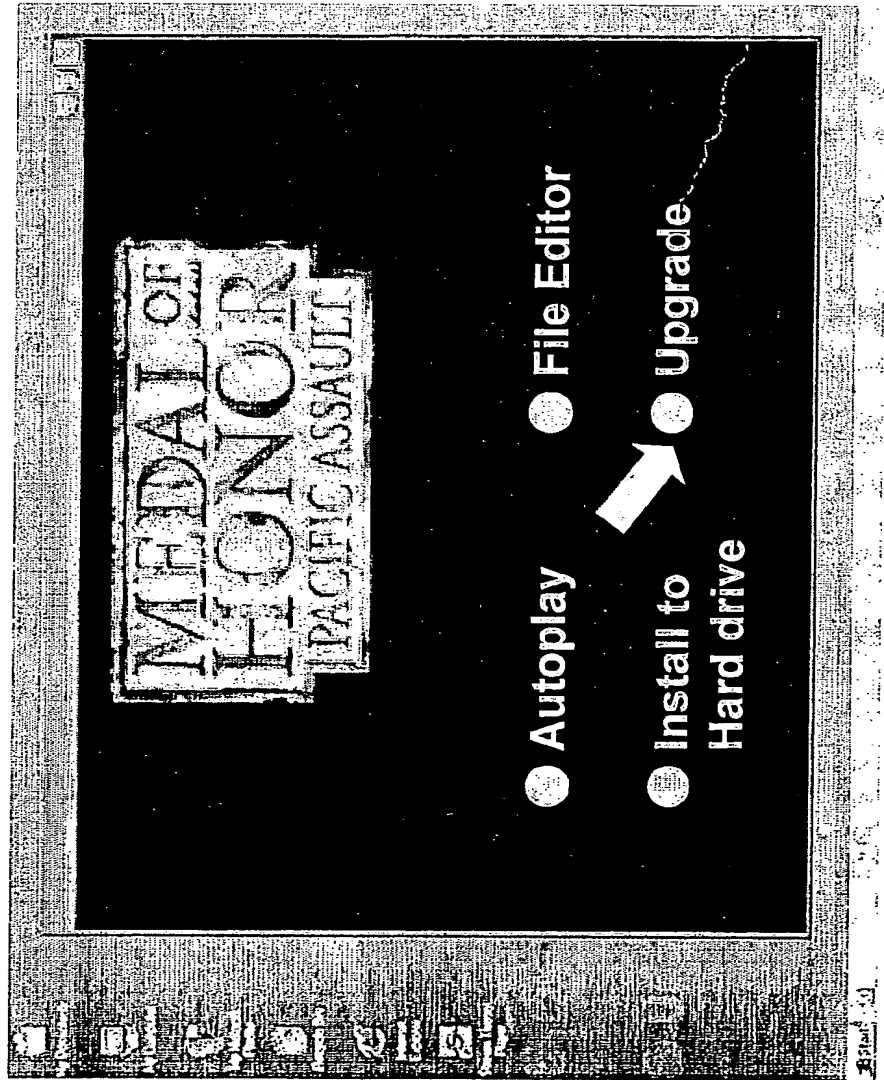
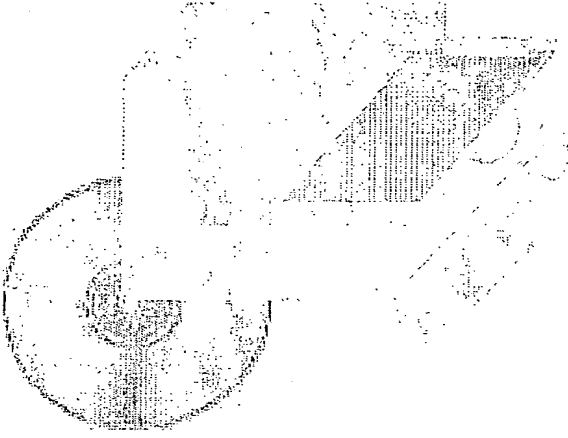
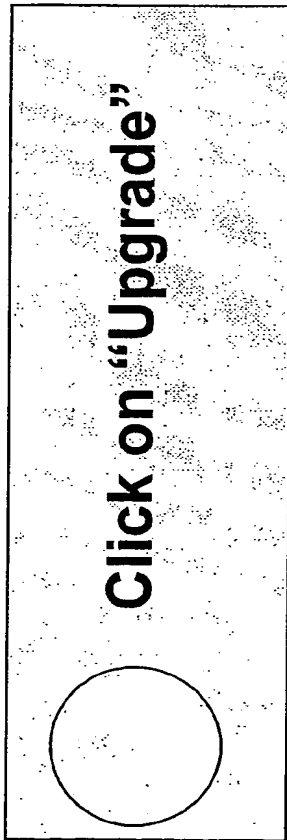


FIG. 2M

The CD Streaming Experience

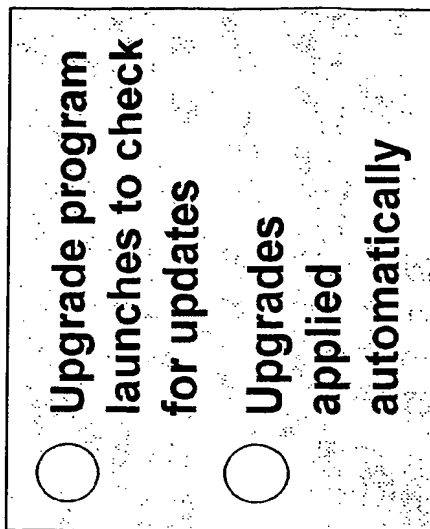
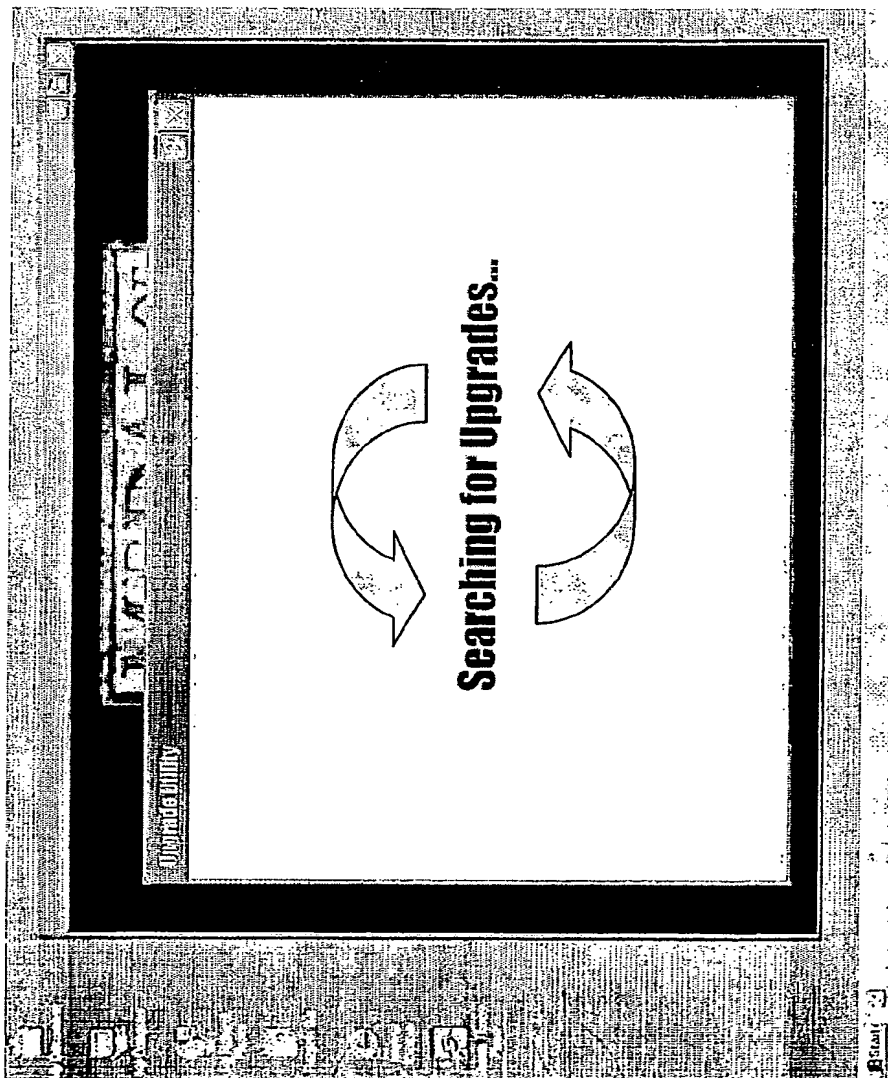
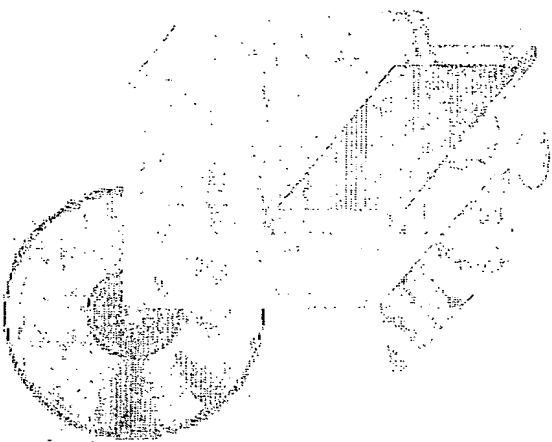
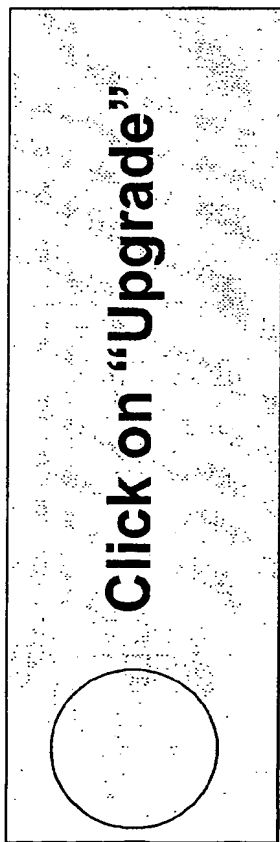


FIG. 2N

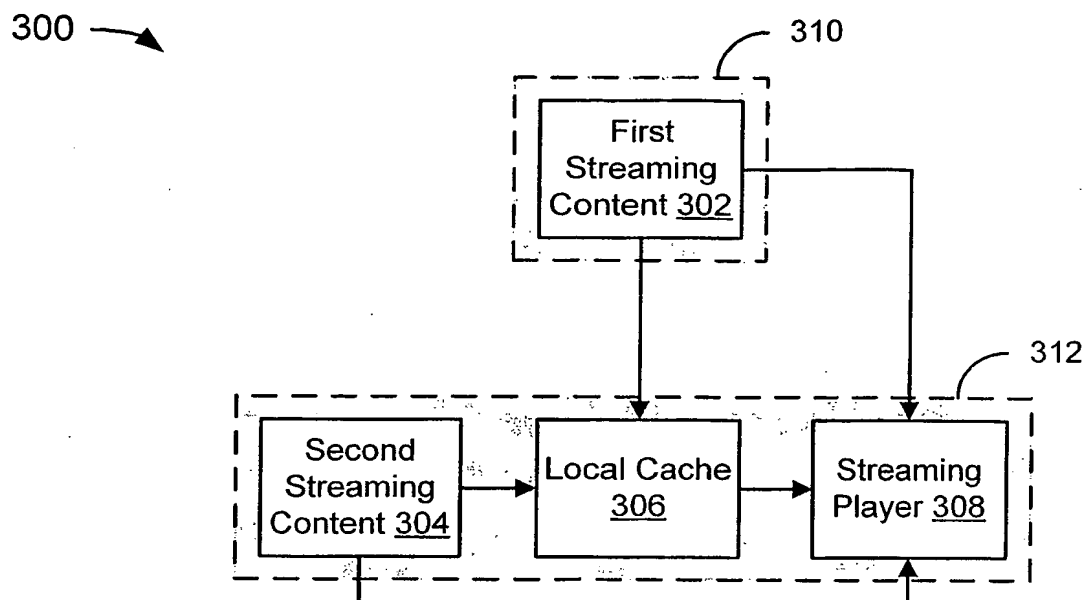


FIG. 3A

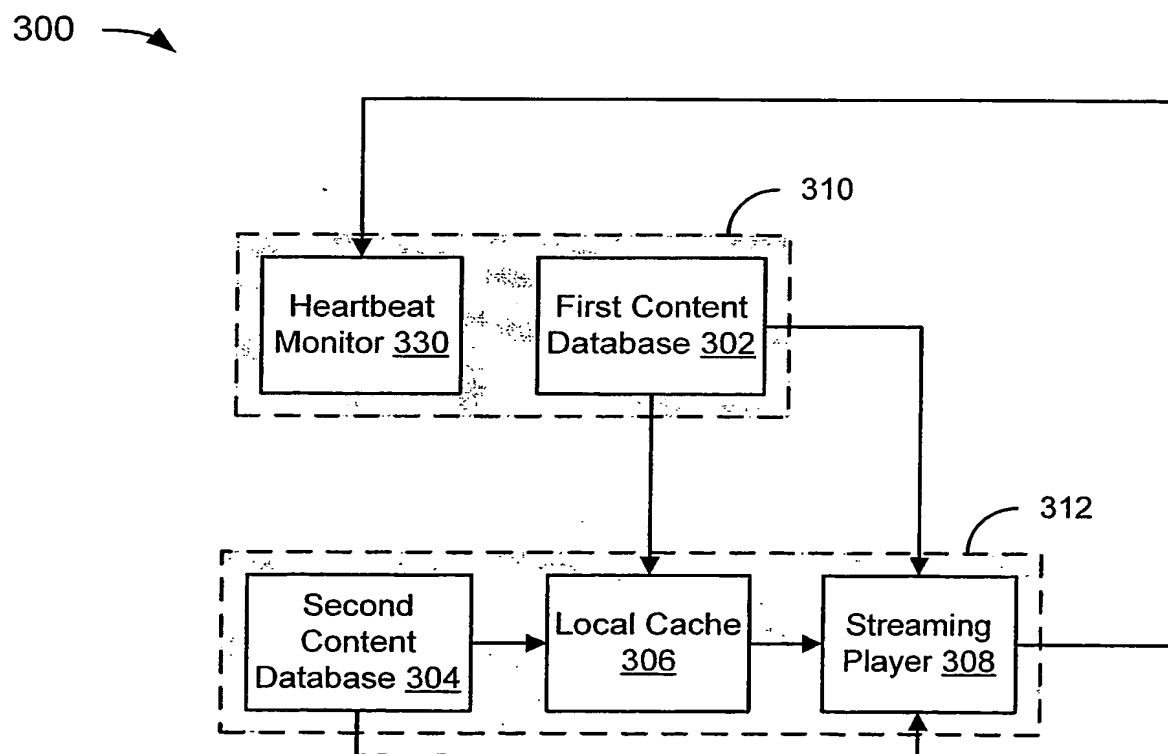


FIG. 3B

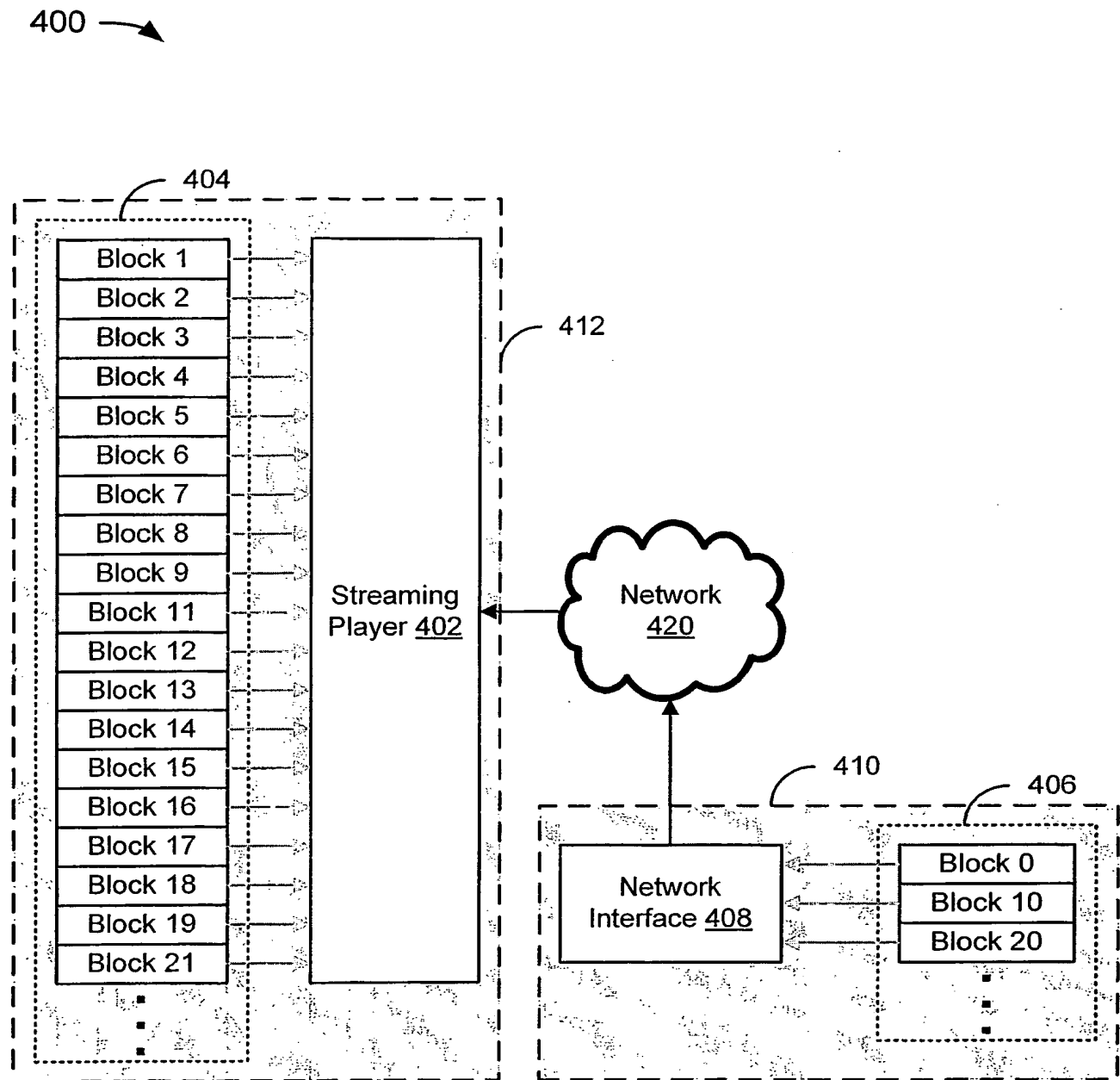


FIG. 4A

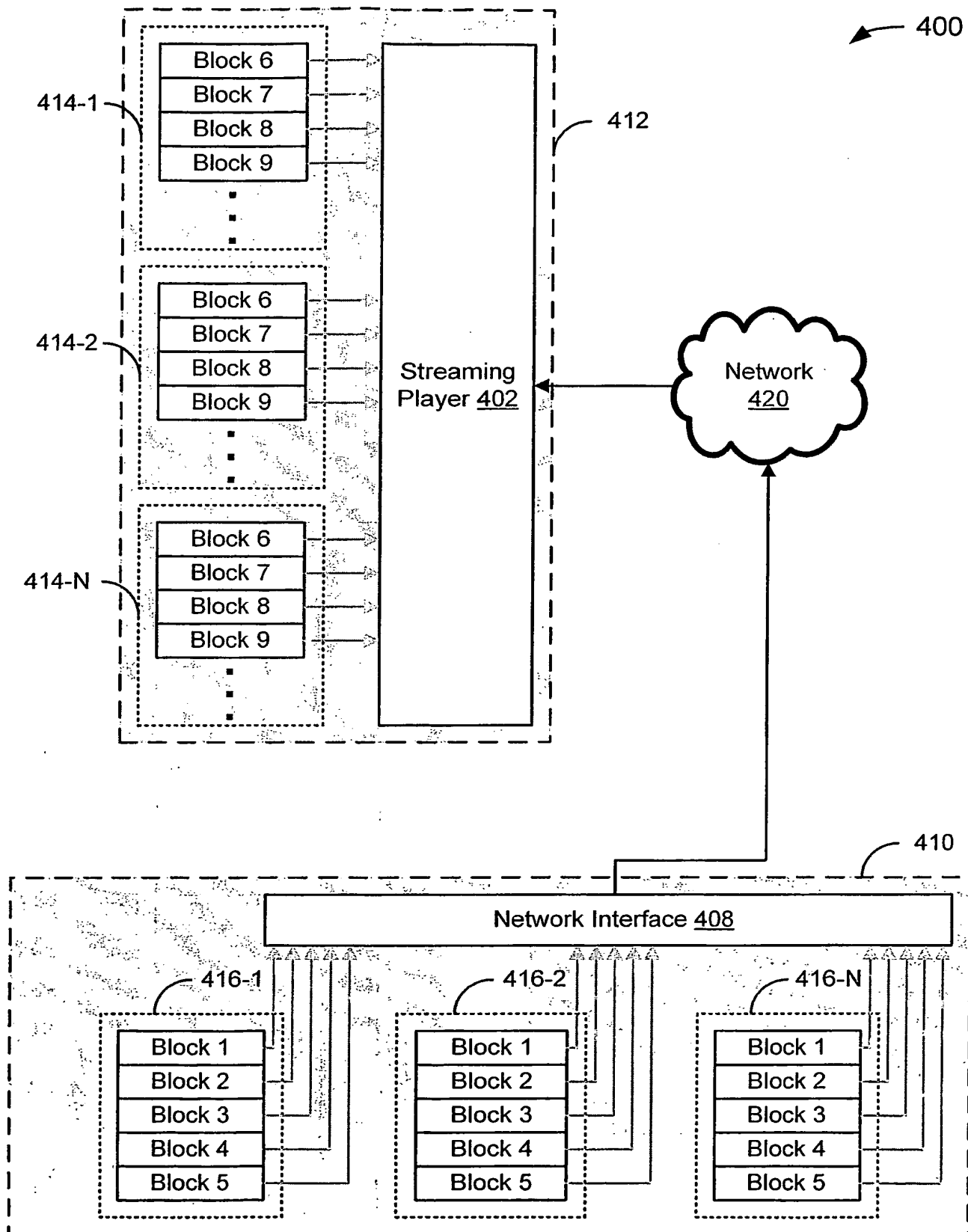


FIG. 4B

500 →

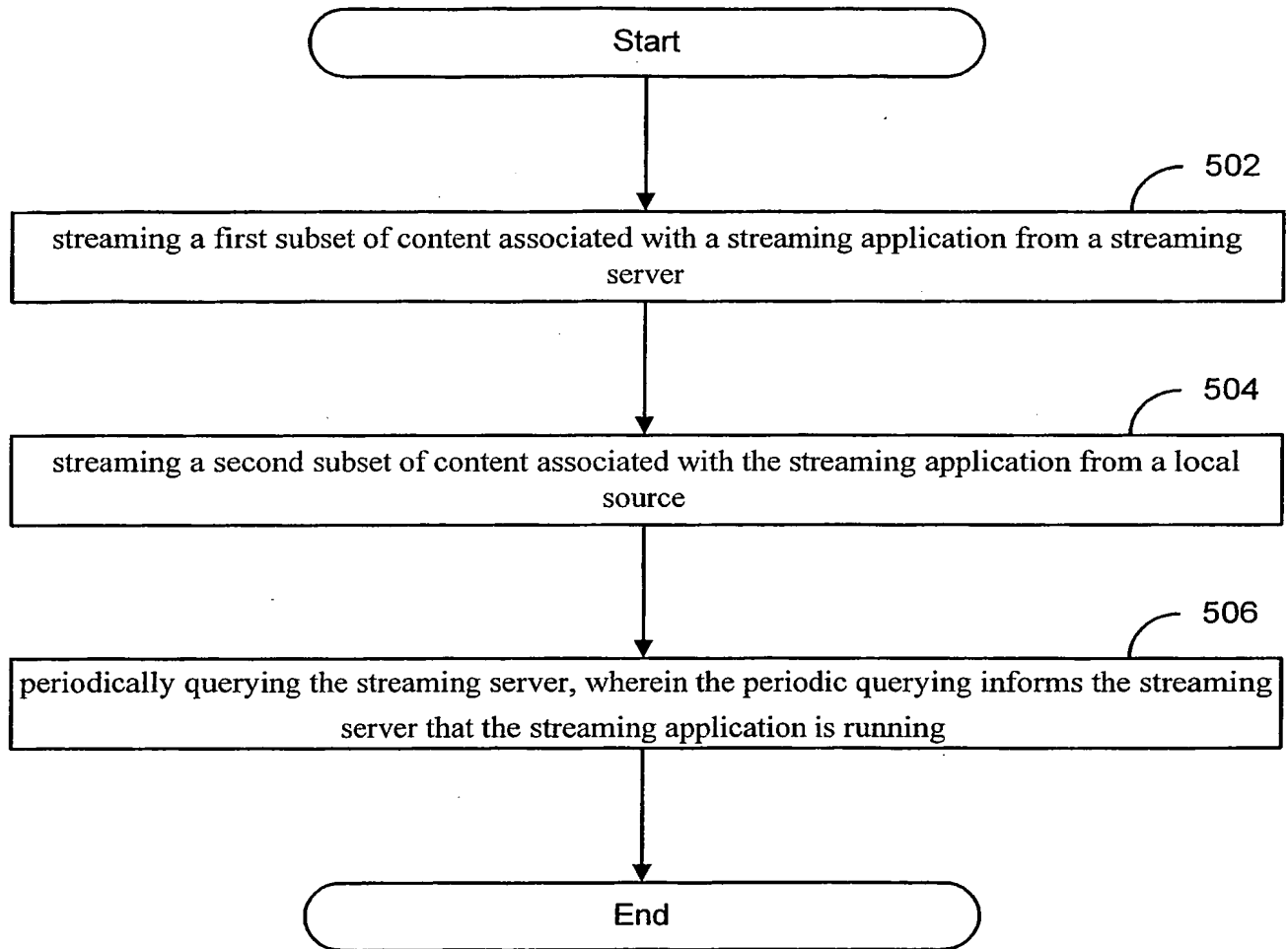


FIG. 5

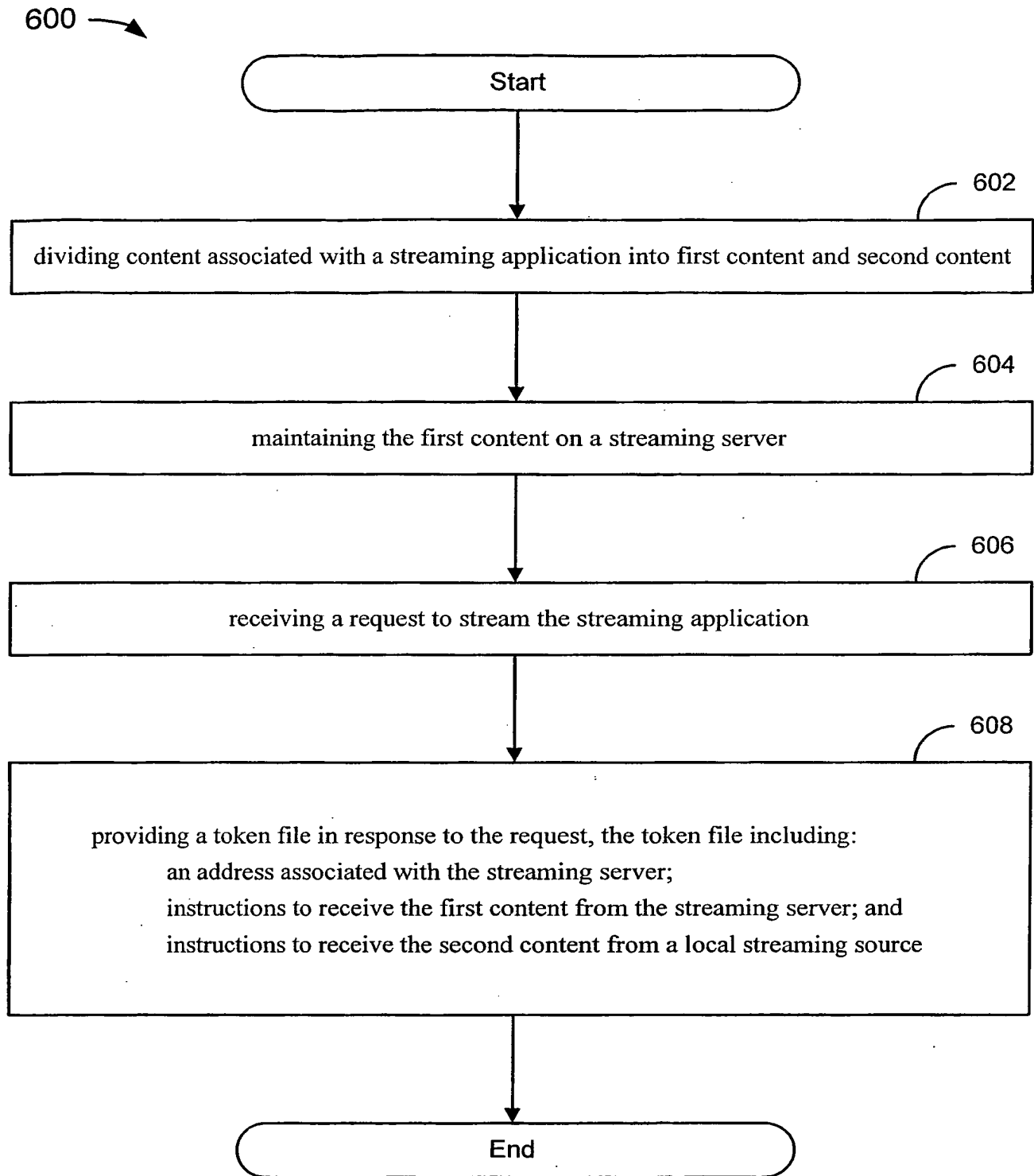


FIG. 6

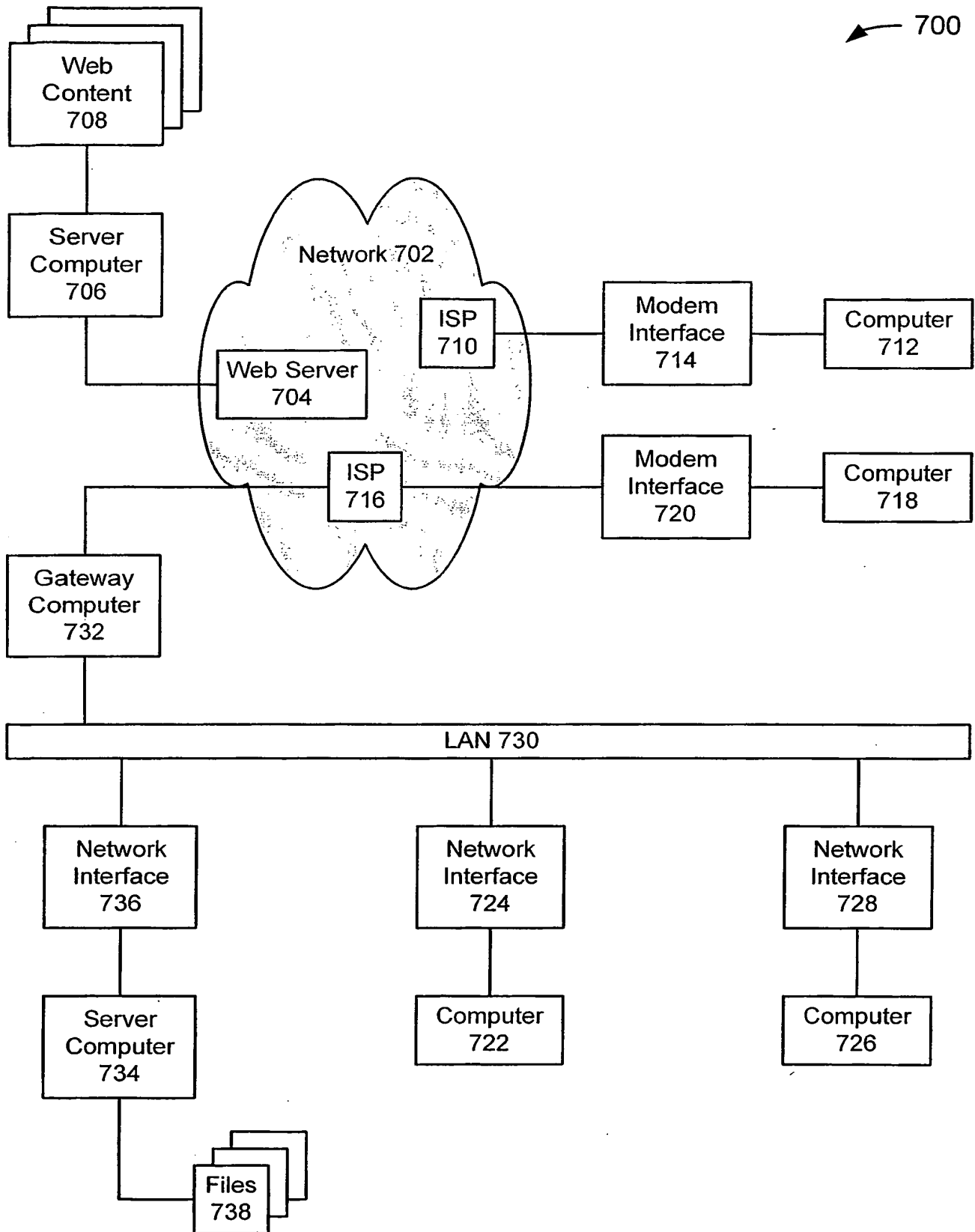


FIG. 7

740 →

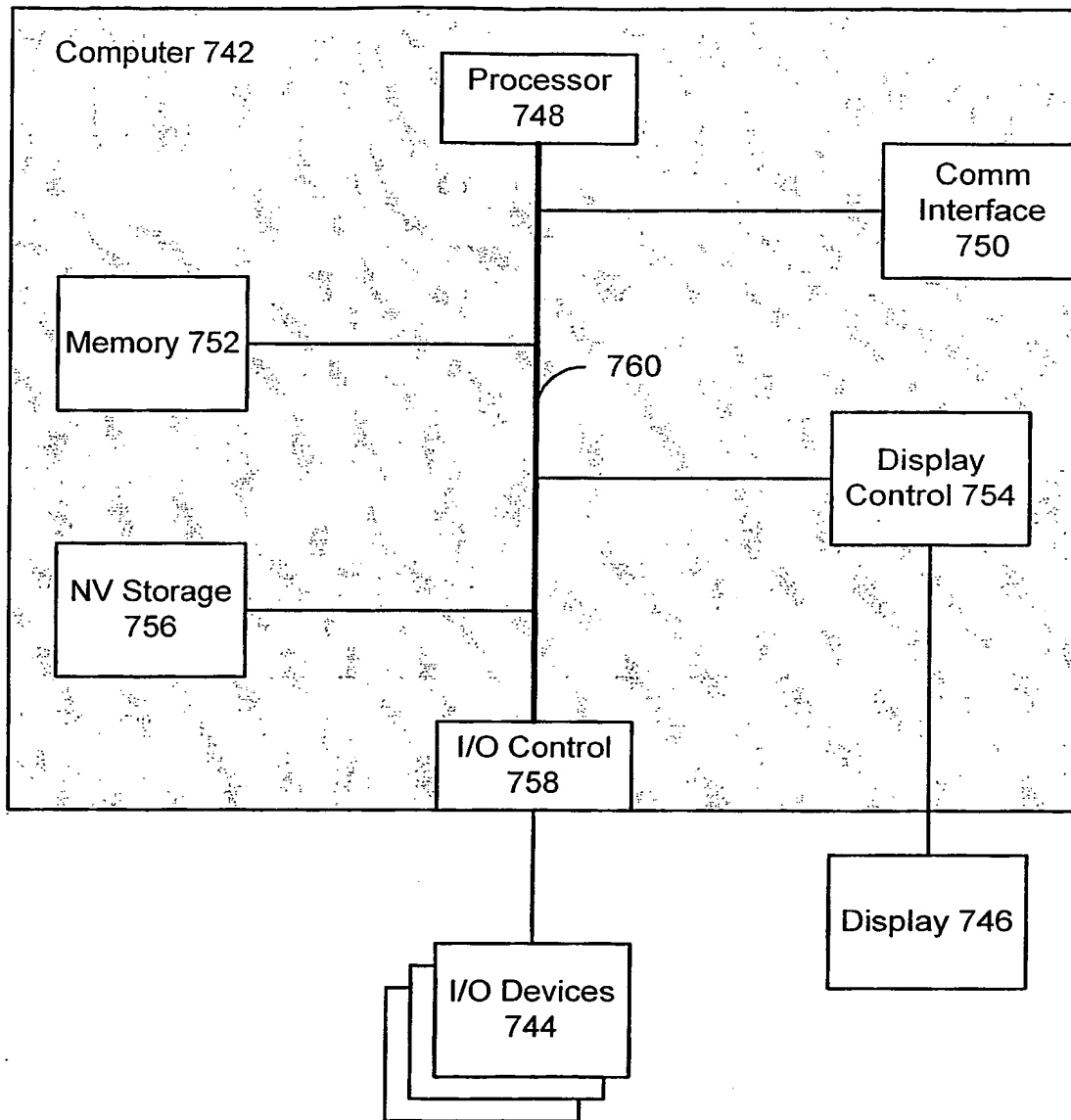


FIG. 8

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.